8 How Volume Management Works

The information in this chapter is intended for the system administrator who wants in-depth knowledge of the volume management software. It describes the volume management processes and how they work together to provide services to EDM Backup and HSM software.

This chapter covers the following topics:

- rvmoper UNIX Group
- Volume Management Processes
- Volume Management Startup
- Library Unit Operations
- Volume Allocation and Deallocation

rvmoper UNIX Group

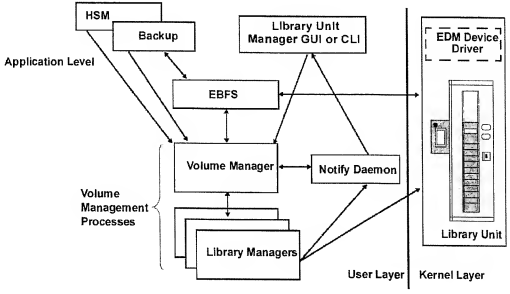
Normally, users who are not root can only monitor volume management activity. However, if you are a member of the rvmoper UNIX group (/etc/group), you can perform volume management functions, such as labeling media, and injecting and ejecting media from a library unit.

Note: To become an rvmoper member, contact your UNIX system administrator, who can add you to the group.

Volume Management Processes

The Volume Management software consists of several independent processes that together provide volume management services to Backup and HSM. Figure 8-1 illustrates these processes:

Figure 8-1 **Volume Management Process Diagram**



You can view currently running volume management processes by using the **evmlistd** command. (Refer to the **evmlistd** man page for more information.) Following is an example of currently running processes:

```
# evmlistd
    root  3966  3964  0   Nov 18 ?        4:36 notd -d
    root  3983  3964  0   Nov 18 ?       40:01 ../atl_3264_0/lmd -
    root  3964     1  1   Nov 18 pts/6   10:18 /usr/epoch/bin/vmd
    root  3984  3964  1   Nov 18 ?        7:05 ../offsite_0/lmd -
    root  3985  3964  1   Nov 18 ?        6:52 ../offline_0/lmd -
Done
```

Volume Manager

The Volume Manager (vmdaemon) is the principal process in volume management. It interacts with EBFS (Epoch Bitfile System), device-specific Library Managers, and the notify daemon. EBFS enables applications, such as Backup and HSM, to write data to removable media. Library Manager daemons (lmd) control library unit operations such as robot movement which transports cartridges to and from an LU's inlet, internal drives, and storage slots. The notify daemon communicates changes between the Volume Manager and Library Managers, and EDM Library Unit Manager graphical user interface (GUI).

You can also view the current status of the Volume Manager by using the **evmstat** command. Various options enable you to view the status of components such as drives, inlets, individual library units, etc. (Refer to the **evmstat** man page for more information.)

Backup and HSM invoke requests to the Volume Manager to open and close volumes, obtain volume status, and to allocate and deallocate volumes.

Library Managers

Individual Library Manager processes manage library unit operations on a per-device basis. A Library Manager is set up for each library unit that is configured for the EDM by using the **lmconfig** utility.

Requests for media are sent from an application to the Volume Manager. Upon receipt of a request, the Volume Manager passes the request to the appropriate Library Manager for processing.

The Library Manager notifies the Volume Manager and EDM GUI, via the notify daemon, when it completes an operation.

Notify Daemon

The Volume Manager and each Library Manager communicates with the GUI by way of the notify daemon (notd). The notify daemon enables the GUI to display up-to-date status and information during system operation. For example, when you label a new piece of media, the Volume Manager sends the new volume label information to the EDM Library Unit Manager by way of the notify daemon. The Media List is updated to display information for the newly labeled volume such as its volume sequence number, the application making the request, and the trail name.

Volume Management Startup

The vmdaemon is started from the system startup file. The vmdaemon starts the notify daemon, sets the parameters defined in its configuration file (vm.cfg), and starts an lmd process for each Library Manager configured for the server.

During startup, each Library Manager daemon reads a unique configuration file (lm.cfg). The configuration file defines the name of the Library Manager, sets the hardware address of the library unit and drives, configures the number of drives and drive features, and sets the library unit's operating and

scheduling parameters. (See "Library Manager Configuration Files" on page C-9 for more information about the parameters in the Library Manager configuration file.)

Manually Stopping and Restarting the vmdaemon

Under normal operating conditions, you do not have to start up or shut down the volume management system. When you reboot the system, volume management processes are automatically shut down and restarted.

However, you may need to shut down and restart volume management if the vmdaemon fails or you determine that one or more processes are in an unknown state. Volume management does not automatically recover from unexpected vmdaemon failures.

Note: Manual shutdown of the vmdaemon should be done only by EMC field service personnel or when instructed by an EMC customer service representative.

Using edmproc -restart

To recover from this type of failure, use the **edmproc -restart** command. This command shuts down the remaining EDM processes and then starts up all of the processes again in the correct order. (Refer to the **edmproc** man page for more information about this command.)

You should be sure no backup, media duplication, HSM, or restore processes are running when you use this command. Use the command **vmdupd -L** to check on media duplication processes, **ebbackup -L** to check on backup processes, **ebrestore** for restore processes, or **emsstat** for HSM processes. (Refer to the appropriate man page for more information about each of these commands.)

Following is an example of the output that appears at the CLI when you use the **edmproc -restart** command:

```
# edmproc -restart

EDM daemon shutdown ...
        Shutting down System Monitoring ... Done
        Shutting down Backup Activity Monitor ... Done
        Shutting down Client daemons ... Done
        Shutting down Backup Server ... 12/27/99 15:10:25
[ 16226:ebcatalogd] Halt signal sent to ebcatalogd process #1830

Halt signal sent to EpochBackup Listener process #1806
Done
        Shutting down Bitfile Services ...    halting ebfsd, process id 356
        Bitfile Services shutdown complete
        Media Duplication shutdown started
        Media Duplication shutdown complete
Done
Status of file /kernel/drv/st.conf is good
    Shutting down Volume Management ... Done
        Shutting down SNMP support ... Done
EDM daemon shutdown complete
EDM daemon startup ...
        Starting SNMP support ... Done
Status of file /kernel/drv/st.conf is good
        Starting Volume Management ... Done
        Starting Bitfile Services ... Done
        Starting Backup Server ... Done
        Startup Client daemons ... Done
        Starting Backup Activity Monitor ... Done
        Starting System Monitoring ...Done
Done

EDM daemon startup complete
```

If an Error Occurs While Using edmproc -restart

If the ebfsd or vmdupd processes do not shut down within one and one-half minutes of the restart request, **edmproc -restart** halts the processes and forces a shutdown. A series of error messages may appear before the shutdown that address one of the following processes: ebfsd, vmdupd, or vmdupmedia (which vmdupd controls).

For example, if a problem occurs with shutting down the ebfsd daemon, the following message appears:

```
Sending kill signal to ebfsd, process id process id
```

If more time passes without a successful shutdown, another message appears:

Process ebfsd pid *pid* is slow to shut down, sending kill signal

If all subsequent attempts to shut down the process are unsuccessful:

```
Unable to shut down (ebfsd) process id process id.
You must perform a UNIX shutdown to terminate this
process.
```

You should then reboot the EDM server to clear this process successfully.

Library Unit Operations

Library Manager daemons handle the following library unit operations:

- Inserting Media into Library Units
- Mounting and Dismounting Volumes
- Ejecting Media from a Library Unit
- Drive Scheduling and Preemption
- Library Unit Inventories

Inserting Media into Library Units

Media cartridges enter a library unit (LU) through the library unit's inlet. Library units have one of two inlet types: automatic or manual. If a library unit has an automatic inlet, the Library Manager polls the inlet periodically for incoming cartridges. If the LU has a manual inlet, you must inform the Library Manager when you place media into the inlet. You do this in the Utilities tab of the EDM GUI's Library Unit Manager window. (Refer to EDM online Help for more information about this window.)

When a Library Manager detects media in an inlet, it first checks the library unit for an available slot. If a slot is available, the robot moves it from the inlet into the next available slot of the LU. This slot becomes the volume's "home slot." The Library Manager inventories the volume and sends information for the new volume to the Volume Manager (by way of the notify daemon). The Volume Manager creates an entry in the volume catalog and notifies the EDM Library Unit Manager to display the new volume in the Media List.

Importing a Volume

Volumes are imported into the EDM system when their status is Uncataloged (refer to Chapter 7 for more information about this volume state). You must inject an uncataloged volume into a library unit before you import it into an EDM.

The import feature is generally used for restore or disaster-recovery purposes, so that you can transfer one or more volumes from one EDM to another. The receiving EDM can then obtain the same information about the volume that the original EDM had.

Refer to Chapter 19 "Recovering a Server from a Disk Failure" and Chapter 20 "Recovering a UNIX Client from Disk Failure" of this manual for information about disaster recovery.

You can import the volume into an EDM through the Library Unit Manager window of the EDM GUI (refer to online Help for instructions). You can also import a volume at the CLI by entering the command **evmimport** (refer to the evmimport man page for more information). For example:

```
# evmimport -V -l atl_452_0 -s 39
```

Using this command adds information about the volume to the volume catalog, and the volume is cataloged.

Importing a Duplicate Volume Before its Original

When importing a duplicate volume into an EDM system where the original volume is unknown, the duplicate's barcode, volume ID, and sequence number are imported with it. The volume ID of the duplicate's original volume is also imported. A "placeholder" sequence number is created for the original volume in the LU. (This placeholder provides the original volume a valid sequence number; it does not affect backup or restore processes.) The original volume's barcode remains blank.

If you then import the original volume into the same EDM system, the original volume's proper sequence number replaces the placeholder sequence number that was created for it. The original volume's barcode also updates to match the real volume.

Gathering Media Information

When a volume is imported, the fields contained on the label (such as volume ID) are set in the Library Unit Manager of the EDM GUI, or by running **evmimport**. Other media information that EDM uses is set by running **ebimport** (such as the ebfs ID and ebfs directory ID).

Other information such as the amount of data written to the media (which appears in the media list information in the Library Unit manager window of the EDM GUI) cannot be retrieved. However, this lack of information does not affect normal EDM operations (backups, restores, duplications, etc.).

Inserting Cleaning Cartridges into Library Units

You insert a cleaning cartridge into a library unit (LU) as you do a data tape. The Library Manager recognizes the tape as a cleaner by its barcode when the cleaner enters the LU. (During configuration of the LU, the default barcode for cleaners is set to CLN*XXX*. Refer to Chapter 17 "Configuring Library Managers" for more information.)

When you inject a cleaner into an LU for the first time, the default for the maximum number of times the cleaner can be used is set to 20. As a cleaning cartridge is used, its remaining uses count is decremented.

Note: If barcodes are not used, the cleaner barcode is not CLN*XXX*, or cleaner barcode recognition is disabled, you must inject the cleaner through the Utilities tab of the Library Unit Manager window, or by using the **evminject -c** command at the CLI (refer to the **evminject** man page for more information).

You can change the maximum uses count by using the **evmchvol** command in which you specify the LU name and slot number where the cleaner resides, as shown:

```
# evmchvol -l library unit name -s slot number maxuses=n
```

If the cleaner was already used a number of times before being inserted into the LU, you can ensure this usage count by using the **evmchvol** command as follows:

```
# evmchvol -l library unit name -s slot number uses=n
```

You can verify the uses or maximum uses count that you set by viewing the values in the Information tab of the Library Unit Manager window, in the EDM GUI.

After the cleaner is used for the first time, the Library Manager tracks the number of times the cleaner is used until it reaches the set maximum.

Note: Be sure to have another cleaning cartridge available when a cleaner in use reaches its maximum usage.

(Refer to the **evmchvol** man page for more information about this command.)

When Drives are Busy

When you insert (inject) media into a library unit, and all drives are busy, the inject does not complete until a drive is available. An inject operation does not preempt any job that is currently using drives, even if that job is a lower priority than the job that requested the inject. (An inject requires a drive to read the volume's label.)

For example, backups have a higher priority than media duplication. But if all drives are busy with a media duplication operation and a request arrives causing the Media Requests window to open, you can insert a volume but the inject does not complete until a drive is available. (See "Drive Scheduling

and Preemption" on page 8-16 for related information.) To avoid this problem, make sure you always have sufficient media in the library unit for the higher priority job.

Inserting media for a higher priority job waits for a lower priority job to release the drive, it does not preempt any job that is currently using the drives, even if that job is of a lower priority than the job that requested the inject. (An inject requires a drive to read the volume's label.)

Note: To avoid this problem, you must make sure you have sufficient media in the library unit for the higher priority job prior to starting the job. You can also configure a trail to use fewer drives during processing (refer to EDM Online Help for more information).

Mounting and Dismounting Volumes

Library Managers handle mount and dismount requests on a priority basis. The following sections describe mounting and dismounting media.

Mounting Volumes

When a mount request arrives, the Library Manager first determines if the volume is mounted in a drive or one of the storage slots. If the volume is already mounted in a drive, the Library Manager sends the Volume Manager the drive number in which the volume is mounted.

If the volume is in one of the library unit's storage slots, the Library Manager schedules the volume for mounting. When a drive becomes available, the Library Manager mounts the volume, reads its volume label, and sends the drive number to the Volume Manager.

If the volume is not in a drive or library unit, the volume is either offline or offsite. If the volume is offline, the Volume Manager generates a request for the GUI to open the Media Request window. The Media Request window displays the volume sequence number and/or barcode ID of the volume

(and its duplicate if one exists). To respond to the mount request, the operator physically locates the volume and inserts it into the specified library unit.

After the volume is inserted into a library unit, the Library Manager schedules it for mounting and the request is removed from the Media Request window. The Library Manager mounts the volume and notifies the application (via the Volume Manager) that the volume is ready for access.

Dismounting Volumes

When a dismount request is received from an application, the Library Manager first acknowledges the request, places the volume in a dismount queue, and leaves it in the drive for a specified number of seconds (LM_MAX_IDLE_TIME determines this number, as configured in the lm.cfg file).

> **Note:** The LM_MAX_IDLE_TIME parameter does not apply when dismounting a volume from the GUI or CLI; in either case, the volume dismounts immediately.

When the time that is specified in lm.cfg elapses, the Library Manager dismounts the volume from the drive and returns it to its home slot. If a request comes in for the same volume during this period of time, the volume is already mounted in the drive and ready for access by the application. The Library Manager avoids remounting the same volume and thereby optimizes drive access.

If the Library Manager receives a mount request for a new volume during a pending dismount and no other drive is available, the volume with the pending dismount status is immediately dismounted to free up the drive so that a new volume can be mounted.

Ejecting Media from a Library Unit

When you eject media from a library unit, the Library Manager schedules the volume(s) for removal from the library unit. Information about each ejected volume moves into the offline Library Manager. If you specify to eject the media to offsite, information moves briefly into the offline LM and then to the offsite LM.

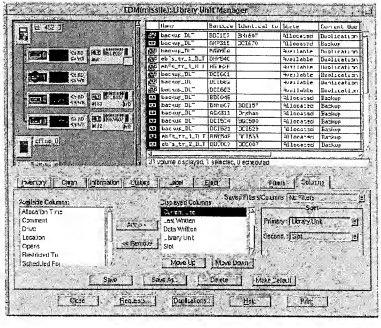
When the Volume Manager receives an eject request, it first checks the catalog for the volume's location and sends a request to the appropriate Library Manager for processing. The Library Manager, upon receipt, schedules the eject request.

If the volume is in a drive at the time the eject request is processed, the Library Manager waits for receipt of a dismount request before ejecting the volume from the library unit.

After the Library Manager ejects a volume, all outstanding requests for the volume are canceled. The Library Manager sends notification to the Volume Manager to close the request; the EDM Library Unit Manager window then reflects the change.

Ejecting Media Through the EDM GUI

You can eject media through the Library Unit Manager window (Eject tab) of the EDM GUI. In this window, select the Eject tab, as shown below. Refer to EDM Online Help for instructions on using this tab.



Ejecting Media at the CLI

At the CLI, use the following command to eject media from a library unit:

```
# evmeject
```

When you issue the command, the prompt does not reappear until the eject operation has completed. However, if you want to run **evmeject** as a background operation, use the command as follows:

```
# evmeject &
```

Refer to the **evmeject** man page for more information.

Drive Scheduling and Preemption

Each Library Manager handles drive scheduling based on a priority that the application establishes. When a Library Manager receives a request (for example, a mount request), the Library Manager adds the request to a prioritized work queue. When a drive becomes available, the Library Manager services the next work order with the highest priority.

Drive Preemption

Drive preemption occurs when a volume is mounted in a drive and an application makes a mount request for a volume with a higher priority. (The Library Manager determines preemption of a volume based on the volume's priority in the queue.) If no other drives are available, the volume with the lower priority is dismounted, which makes the drive available to the volume with the higher priority.

Verifying Priority in the Queue

An application, with a mounted and open volume, periodically polls the Volume Manager to verify whether the volume should be removed from the drive. The Volume Manager, in turn, asks the Library Manager to check for mounts of volumes with a higher priority. If a volume with a higher priority is waiting to be mounted, the Library Manager notifies the application by way of the Volume Manager to close the volume in the drive. After the application closes the drive, the Library Manager dismounts the volume, making the drive available to the volume with higher priority.

If all application requests are of equal priority, they are scheduled on a round-robin basis. For example, if five volumes have the same priority and only one drive is available, each application gets a time slice of the drive.

LM_MAX_RESIDENT_TIME and LM_MIN_RESIDENT_TIME

Two parameters essentially govern drive usage: LM_MAX_RESIDENT_TIME and LM_MIN_RESIDENT_TIME.

- LM_MAX_RESIDENT_TIME is the maximum time (in seconds) that a volume can remain in a drive before allowing preemption by a volume of the same priority. The default value for tape drives is 7200 seconds (two hours); the default value for EO drives is 120 seconds (two minutes).
- LM_MIN_RESIDENT_TIME is the minimum time (in seconds) that a volume with a lower priority can be in a drive before allowing preemption for a volume of a higher priority. The default value for tape drives is 120 seconds (two minutes); the default value for EO drives is 30 seconds.

Note: You can modify both parameters in the lm.cfg file. Refer to Appendix C "Volume Management Configuration Files" for more information about lm.cfg.

Note: Though you can set LM_MIN_RESIDENT_TIME to any value, the EDM applications (backup, restore, HSM, and media duplication) poll drive usage every five minutes. Therefore, if you set the value to less than five minutes, preemption does not occur until five minutes pass and the application verifies whether preemption is necessary.

Simultaneous Backup Example

The following example describes the process of two simultaneous backups that are contending for the same drive.

1. The first backup process sends a request to the Volume Manager to open and mount volume sequence number 42.
2. The Volume Manager checks the volume location in the volume catalog and sends a request to the appropriate Library Manager to mount volume sequence number 42.
3. The Library Manager creates a work order to mount volume sequence number 42.

4. A second backup process sends a request to the Volume Manager to open and mount volume sequence number 123.
5. The Volume Manager checks the volume location in the volume catalog and sends a request to the appropriate Library Manager to mount volume sequence number 123.
6. The Library Manager creates another work order to mount volume sequence number 123.
7. Periodically, the Library Manager is asked if the volume in the drive should be dismounted.
8. The Library Manager makes a decision based on a volume's priority. If volume sequence number 123 has a higher priority, the Library Manager removes volume sequence number 42 from the drive and mounts volume sequence number 123 (if volume number 42 was in the drive at least for the time specified in LM_MIN_RESIDENT_TIME).

   If both volumes have equal priority, the drive is shared until one of the backup processes finish, or until the time specified in LM_MAX_RESIDENT_TIME has passed.

Library Unit Inventories

A Library Manager inventories the contents of a library unit based on the type and verification criteria you select from the Inventory tab of the GUI. Inventory type enables you to inventory the entire contents (all slots) or only a selected portion of a library unit.

Note: A label and barcode inventory is recommended.

For most library units, a Library Manager inventories each volume that enters a library unit by way of the inlet. If your library unit does not have an inlet, you need to update the Library Manager anytime you change the contents of the library unit by running an inventory. You must also inventory a library unit anytime you bypass normal Library Manager operations (for example, you move a volume using the library unit's front

panel switches). The type of inventory you do depends on the activity or change you make inside the library unit after opening the door.

If you open a library unit door, manually move media, and then shut the door, the status of the volumes in the library unit becomes unknown. To recover from this state, you need to run a label and barcode inventory so that the Library Manager knows what volume is in each slot. The type of inventory you perform is based on what changes (if any) you make inside the library unit after opening the door.

Note: If your library unit has an inlet, lock the front access door and always use the inlet to insert and remove tape cartridges. If you need to open the access door to insert and remove large amounts of media, be sure to perform a delta inventory after each move.

Inventory Tables

Each Library Manager maintains an internal inventory table in the appropriate Library Manager subdirectory in /usr/epoch/etc/lm/volid.dat.

Note: The volid.dat inventory file is created when a Library Manager is initially started by the vmdaemon. This file is used exclusively by the Library Manager and must not be deleted.

The table includes a volume ID, barcode label (if configured in lm.cfg), slot number, and drive location for each volume in the library unit. During an inventory, the Library Manager compares the slot contents of the library unit with the information in the inventory table. If any discrepancies are found, the Library Manager updates its table and sends the changes to the Volume Manager for cataloging.

How Inventories are Done

When a Library Manager receives a request for an inventory, it begins the process by notifying the EDM Library Unit Manager that an inventory is in progress. The word "inventory" appears next to the appropriate library unit in the Library Unit and Drives area of the GUI. The Library Manager creates a list of slots (based on the type you select) to inventory.

As the Library Manager inventories each volume in its list, the slot contents are updated in the inventory table. If it detects any changes, the Library Manager sends the slot contents to the Volume Manager for cataloging.

You can schedule an inventory during normal system operations. An inventory always has a low priority to ensure that maximum system performance is maintained. The Library Manager processes incoming requests of a higher priority, such as mount and dismount requests, before handling an inventory. If a mount request comes in for a volume that is already in the inventory queue, the Library Manager inventories the volume and removes it from the queue.

After the Library Manager inventories the last item in the list, it informs the Volume Manager and the notify daemon that it completed the inventory. The Volume Manager modifies the volume catalog based on the changes that the Library Manager provided and the GUI is updated.

During the inventory process, the Library Manager processes incoming operations that are of a higher priority. Therefore, a full library unit inventory by label does not have a noticeable effect on overall system performance.

Delta Inventory

When you perform a delta inventory, the Library Manager:

- checks each slot in the library unit to determine which slots changed.
  - If the slot was full and is now empty, the Library Manager marks the slot as empty and removes the volume from its inventory table.
  - If the slot was empty and is now full, the Library Manager marks the slot as needing verification.
  - If the barcode of a volume in a slot is different from the barcode of a volume that was previously in the slot, the Library Manager marks the slot as needing verification.
  - If no change was made to the slot, the Library Manager skips the slot.
- creates a work order for each slot that needs verification and inventories each item using the specified verification criteria.

Barcode Inventories

If a library unit is equipped and configured (in lm.cfg) to read barcode labels, you can verify only the barcode label or you can verify both the barcode label and the volume label.

The Library Manager does a barcode inventory by scanning the barcode label of each volume in the inventory queue. Because no volume mounting is involved, a barcode inventory takes significantly less time to complete than a label inventory. The Library Manager updates the barcode ID for each volume in its inventory table and informs the Volume Manager of any changes.

When you select Verify Both Label and Barcode from the Inventory tab of the EDM Library Unit Manager window, both the barcode label and volume label are verified for the selected inventory type. (Refer to Help for the Inventory tab for details.)

Note: It is recommended that, when volumes are regularly rearranged in a library unit through a mechanism other than the documented inject and eject operations (for example, insertion and removal through the library unit's mass-load door), you run a complete label and barcode inventory rather than the simple barcode inventory. This averts volume barcode and ID mismatch and other related problems that could arise in the volume catalog.

Note: Do not use duplicate barcodes. Attempting to add duplicate barcoded media to the system causes unpredictable results.

Cleaning Tape Drives

The EDM software detects when drives need to be cleaned. If a cleaning cartridge is loaded in the library unit, EDM mounts the cleaning cartridge and cleans the drive automatically.

The **evmclean** command (which you can add to a cron procedure) requests cleaning for specific drives in a library unit. The drives are cleaned when they become available. **evmclean** cleans as many of the indicated drives as possible before exhausting the uses that remain on the cleaning cartridge. (Refer to the **evmclean** man page for more information.)

Note: For procedures about cleaning drives manually in the EDM GUI, refer to EDM Online Help.

At least one cleaning cartridge with remaining uses should always be available in each library unit. If a drive requires cleaning and a cleaning cartridge with remaining uses is not available in the library unit, the drive is disabled until it is cleaned. Any data tape that is mounted in the drive when it goes dirty is dismounted. If, at that time, no unused drives are available in the library unit, the performance of any active backup, duplication, restore or HSM operation may be negatively impacted.

Volume Allocation and Deallocation

The process of volume allocation and deallocation is initiated at the application level. When an application needs a volume allocated to a trail, it sends a request to the Volume Manager.

An application determines when data on a volume is no longer needed and is ready for deallocation. Once a volume is deallocated, it becomes available for allocation. Backup determines if a volume is ready for deallocation when expiring backups. HSM handles volume deallocation following compaction.

How Volumes are Allocated

During a backup, when the current volume is filled, the application makes a request for another volume in the same trail. The Volume Manager searches its drives and library units for an available volume that matches the request. If one is available, it is allocated to the application.

Additional volumes become available to a trail when you label media. You specify the trail name that can use it by choosing the appropriate volume template. You can also specify whether the volume is part of an available pool of new volumes, or it is available for a specific trail name.

If no available volumes are in the library unit, the Media Request window alerts the operator to make a new volume available. The window includes any prelabeled volumes that are offline and any unlabeled volumes that can be labeled for the request.

Volume Allocation Request

When an application needs a new volume, it sends a volume allocation request to the Volume Manager. The request includes a template that defines the type of volume it needs.

Upon receipt of the request, the Volume Manager creates an entry in the volume catalog, assigns a volume ID to the request, and sends the volume ID to the application. The application retains the volume ID. No physical media is associated with the request at this time; therefore, a volume sequence number is zero or barcode ID does not yet exist for the volume.

When the application is ready to use the volume, it sends an open request and a mount request with the volume ID to the Volume Manager. The Volume Manager searches the volume catalog for a suitable volume. When found, it sends a mount request to the appropriate Library Manager.

Mount Request

When the Library Manager receives the mount request, it schedules the request, mounts the volume when a drive is available, and relabels the volume as Allocated. The Library Manager notifies the Volume Manager when the volume is ready for the application to access the volume.

Note: If no available volumes match the request, the Volume Manager sends a volume allocation request notification to the EDM Library Unit Manager. The Media Request window opens and displays NEW in the Sequence # field.

Volume Use

The application retains the volume ID and continues to use the volume until it completes writing to the volume or the volume fills up (for example, a tape reaches the end of the media). When the application fills one volume and needs another, it sends another volume allocation request to the Volume Manager and the process repeats.

When the Volume Manager looks for a suitable volume, it looks for a volume that meets these requirements:

- The volume is not currently allocated to another application.
- The media type matches the media type that is specified in the template.
- The volume did not exceed its maximum use count. If the maximum use count is set to 0, its use is unlimited.
- The volume restriction is reviewed (volume restrictions for a given media set are defined when the backup is configured):
  - The volume is labeled as Unrestricted, which allows the volume to be allocated to *any* requesting application.
  - The volume is labeled as Restricted by Application which means the volume can be allocated only to the specified application.
  - The volume is Restricted by Name which means the volume can be allocated only if the trail name is the same as that specified by the requesting application. This ensures that once a volume is allocated to a specific trail, it cannot be reallocated to a different trail.

Duplicate Volume Sequence Numbers

Normally, volume management does not allow two volumes of the same media type to share the same volume sequence number. If you attempt to import a volume with a sequence number that already exists on the server, you are prompted to either cancel the import operation or to delete the existing volume that has the same sequence number.

However, if your site has existing volumes with duplicate sequence numbers and you want to import both (or all) volumes, you can override this restriction. The imported volumes then have the same sequence numbers but different barcodes and volume IDs.

Using duplicate numbers does not affect the running system. The sequence numbers enable you to identify individual volumes and do not affect the system's operation.

To allow importing of duplicate sequence numbers, change the value of VM_ALLOW_DUP_SEQ_IMPORT (in the file /usr/epoch/etc/vm/vm.cfg) to "yes."

> **Note:** After you change this value to "yes," no warning is given when you import volumes with duplicate sequence numbers.

Following are the steps to incorporate changes to the vm.cfg file after adding support to allow duplicate sequence numbers:

1. Obtain the process ID (pid) of the vmdaemon:

   ```
   # evmlistd

   root   382     1  0 12:00:29 ?          2:26
   /usr/epoch/bin/vmdaemon -d
   ```

2. Send a HUP signal to the vmdaemon to pick up any changes to the vm.cfg file:

   ```
   # kill -HUP vmdaemon_pid
   ```

   For example:

   ```
   # kill -HUP 382
   ```

When Volumes are Deallocated

When an application deallocates a volume, the Volume Manager takes action based on the volume's media type.

- If the media is DLT, HITC, or DTF, the volume state changes to Available. When a tape cartridge reaches its maximum usage, the volume state changes to Expired.
- If the media is EO, and is configured for auto-erase, the Volume Manager erases the volume and changes its state to Available.

  You can relabel a two-sided optical disk only when both sides become available. The Volume Manager treats all other state changes on an individual side basis. Thus, one side of the disk can be allocated without requiring that the other side be allocated; one side of the optical disk can be expired, erased, made available, and allocated without affecting the other side.
- If the media is a WORM (Write Once, Read Many) optical disk, its data cannot be erased. The state of a WORM optical disk starts as unlabeled. After it is labeled, the disk has a life cycle of Available, Allocated, and then Expired.

For more information about volume states, see "Volume Life Cycle" on page 7-4.

9 Media Duplication

Media duplication enables you to create a duplicate set of backup media automatically after each backup session. You can then use the duplicate set for disaster recovery purposes. This chapter includes the following topics:

- The Media Duplication Process
- Starting Duplication
- Determining Duplicates of an Original Volume
- Manually Disabling and Re-enabling Duplication
- Pausing, Resuming, Canceling, or Removing a Duplication
- Restoring from Backup or Duplicate
- If a Duplication Fails
- Restoring from Backup or Duplicate
- Viewing Reports on Duplications
- Importing a Duplicate Volume
- Rejecting a Mount Request
- Expiring a Duplicate Volume

The Media Duplication Process

Media duplication enables you to make a duplicate set of tape media automatically after a regular backup session. You can then send the original volume offsite and keep the duplicate copy onsite for restore purposes. (Either an original or duplicate may be used for restore purposes.)

A duplicate set of media is of the same media type and has the same theoretical size as the original media. Duplicate media are also uniquely labeled; that is, a duplicate volume has a different volume ID than its corresponding original.

Duplication of backup media can occur automatically after each backup session (unless a trail is set for manual duplication). All media that backup creates can be scheduled for duplication.

Note: You cannot duplicate volumes that are used by Hierarchical Storage Management (HSM).

Duplication of a trail starts after its backup completes. However, if backup of a trail requires more than one volume, neither is duplicated until the backup of the entire trail completes.

Media duplication can run when an unrelated backup or restore process is running, although backup or restore processes take precedence. If a backup or restore activity starts during media duplication and only one drive is still available, both the original and duplicate in a current duplication process share that drive. If no other drives are available, duplication is suspended until an operation with higher priority completes.

Older-generation duplicate volumes are purged automatically when a trail is configured to use new mode for duplication. (Refer to "Starting Duplication" on page 9-3.) These older duplicate volumes are purged when the current duplication of the original volume completes successfully. Only one allocated duplicate volume exists for an original volume.

Note: The automatic purge feature cannot be disabled.

Media Duplication Commands

To manage and control media duplication processes, you can manage the vmdupd, and vmdup daemons, and use the **vmdupcfg** command. The following briefly describes each of these; examples of using each are provided later in this chapter. (Refer to the appropriate man page for detailed information.)

The vmdupd daemon manages media duplication. This daemon, which runs in the background, is part of the normal EDM startup process. vmdupd monitors online tape volumes that require duplication and starts the vmdupmedia processes that perform duplication of those volumes. When run from the command line, you can monitor and control vmdupd's actions.

The vmdup daemon command manages the media duplication scheduling queue. You use this command to start or stop duplication for a trail that is configured for manual duplication. You can also remove duplications from the schedule queue before they begin duplication, or reschedule failed duplications.

Using the **vmdupcfg** command enables you to view the current state of duplication parameters, or modify particular parameters. This command also allows you to enable or disable media duplication system wide.

Starting Duplication

When starting media duplication, you prepare a backup volume for duplication, configure the duplication in append or new mode, configure a backup trail for manual or automatic duplication, and initiate duplication.

The following sections describe each of these procedures.

Preparing a Backup Volume for Duplication

At the beginning of the backup process, a padded tape label (a tape label and padding block) is written to an *original* volume, even if duplication is disabled. This helps to ensure that all of the original data fits on the duplicate media if the duplicate is found to have many bad blocks. Thus, the actual data that is written to the duplicate is exactly the same as that on the original volume. That is, there is always a one-to-one correspondence between the original volume and the duplicate. (The current default pad block is 10 Mb for all tape media.)

You can control the size of the padding block of an original volume at the time a volume is created. For example, you may want to change the pad size on new volumes if your volumes tend to have a high number of bad blocks. You specify the padding block size at the command line using the **vmdupcfg** command with the optional **-tape_pad** argument as follows:

```
# vmdupcfg -set -tape_pad n
```

where ***n*** equals the number of blocks. The value of ***n*** can range from 1 to 50, where 1 = 10 Mb and 50 = 500 Mb; the default value is 10.

> **Note:** The new tape pad size takes effect the next time a backup process allocates new media.

For example, if you set the number of blocks to 4:

```
# vmdupcfg -set -tape_pad 4
```

You can check the setting with vmdupcfg as shown:

```
# vmdupcfg
Media duplication enabled.
No automatic media ejection from the library unit(s).
Tape pad blocks: 4
Max concurrent dups: 1
```

Selecting Append Mode or New Mode

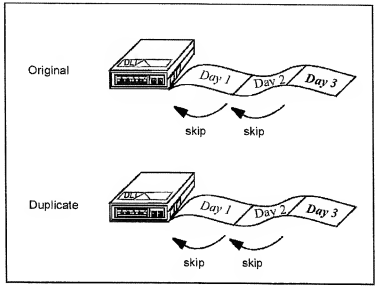
You can configure media duplication for append mode or new mode, as described below.

Append Mode

In append mode (the default), duplication follows the policy that the backup procedure uses. That is, if the backup appends, or adds, data to an existing volume or writes data to a new volume, duplication also appends to an existing volume or uses a new volume, respectively.

When appending to an existing duplicate volume for a trail, duplication starts from the beginning of the volume but skips the previous backup duplications that correspond to the original backup volume. It then adds new backup information to the end, as shown in Figure 9-1.

Figure 9-1 **Append Mode: add data for Day 3 after Day 2.**



In the above example, backup of a trail for Day 3 is appended to the original volume; the data for Day 1 and Day 2 is skipped and Day 3 is added. Duplication of this original volume follows the same policy as that for the original; data for Day 3 is appended after that of Day 2 on the duplicate volume.

New Mode

When you use new mode for duplication, a new volume is allocated for the duplication even if the backup procedure calls for appending to its existing backup volume. Using new mode duplicates the entire tape each time, from beginning to end.

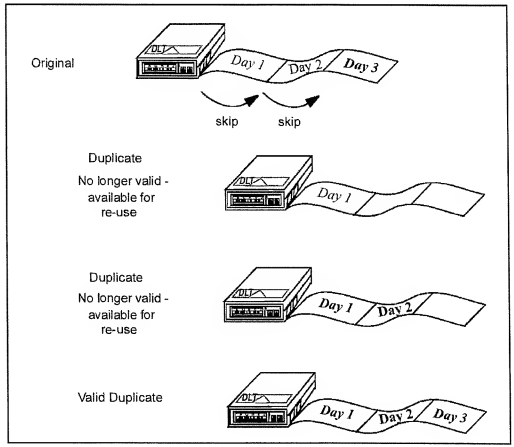
In Figure 9-2, data for Days 1, 2, and 3 of a trail's backup exist on one original volume. During duplication in new mode, data for Day 1 is written on one duplicate volume, data for Days 1 and 2 is written on another, and data for Days 1, 2, and 3 is written on yet another. The duplicate volumes with data for Day 1 and Days 1 and 2 are now considered older generation volumes and are no longer valid. As each duplication completes successfully, the previous, older duplicate volume is reallocated for re-use.

> **Note:** It is **strongly** recommended that you use append mode for duplications. When using new mode, the time to duplicate the original volume increases as more backups are appended to it.

If it is necessary to use new mode, select the Media tab in the Backup Configuration window. In the Duplicate Media section of the window, select the button that is labeled Use New Media for Duplication.

> **Note:** Be sure that new media is available for use in the library unit.

Figure 9-2 **Using New Mode for Duplication.**



Configuring a Trail for Duplication

You configure a backup trail for automatic or manual duplication in the Backup Configuration window of the EDM GUI. In this window, you select the Media tab and then click on the Duplicate Media button.

Selecting automatic duplication enables automatic duplication of a trail when a backup completes. In the mounted volume display area of the Library Unit Manager (LM) window, an application ID of "MD" on the mounted original and duplicate volumes indicates that duplication is in progress.

Note: It is recommended that you use automatic duplication (the default) when duplication is enabled.

Selecting manual duplication in the Backup Configuration window enables duplication of a trail if you wish to duplicate the media at a later time (refer to "Initiating Manual Duplication" below).

Note: Refer to "Backup Trailsets" on page B-58 for an example of a trail that is configured for duplication.

Initiating Manual Duplication at the CLI

You initiate manual duplication of an original volume by trail name at the command line by using the **vmdup** command with the -set and -trail options, and the trail that you wish to duplicate:

```
# vmdup -set -trail trailname
```

An example with output follows:

```
# vmdup -set -trail backup_DLT
Trail backup_DLT scheduled for duplication
```

You can also initiate duplication of an original volume by trail ID by using the -trailID option and the trail ID. To obtain a trail ID, run **ebreport media**; a trail ID appears as a rotation ID in the listing.

```
# vmdup -set -trailID trailID
```

An example with output follows:

```
# vmdup -set -trailID 82D82B05.A9730CA5.00000200.380491BA

TrailId 82D82B05.A9730CA5.00000200.380491BA
scheduled for duplication
```

Note: Ensure that media is available online for duplicate volumes.

Note: Ensure that the original volume is *online* (not offline or offsite) before initiating duplication.

You can use the command **vmdupd -all** to verify that duplication was scheduled. This command displays all active, suspended, and scheduled duplications. (See also "Verifying the Status of a Duplication" below.)

An example follows:

```
# vmdupd -all
Media duplication daemon started.

Active Duplications

Trail: backup_DLT  Mode: append Type: DLT tape  Status: ACTIVE
Orig Vol: A1D7F9BD71812B77 (BDE098)  Seq #: 000025 in TLU: atl_3264_0
Dup  Vol: 45D81F75C195EEF0 (ASV891)  Seq #: 000027 in TLU: atl_452_0
40% Complete  Duration: 000 Hrs. 53 Min. Start Time: 12/31/1999 13:27:16
Blocks Copied: 141381  Total Blocks to Copy: 350343

No Suspended Duplications found

No Scheduled Duplications found
```

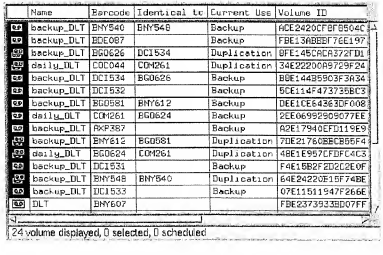
Note: A completed duplication may still appear as Active with a completion status of 100% when you check its status by using **vmdupd -all**. This duplication is still transitioning from an Active state to the duplication state of Done.

Determining Duplicates of an Original Volume

In the Library Unit Manager window of the EDM GUI, you can determine whether a duplicate exists for an original backup volume. In the media list (as shown below), you can identify a duplicate in a number of ways:

- The media list icon in the first column indicates a duplicate by a double tape graphic
- "Duplication" appears in the Current Use column if the volume is a duplicate
- In the Identical To column, the barcode for a duplicate volume appears with its original, and vice versa

  Note: Refer to the Library Manager Online Help (Columns tab) for instructions on adding columns to the media list.



	Name	Barcode	Identical to	Current Use	Volume ID
	backup_DLT	BNY540	BNY548	Backup	6CE2420CF8F8504C
	backup_DLT	BDE087		Backup	FBE13ABBBF7EE197
	backup_DLT	BGD626	DC1534	Duplication	0FE145CACA372FD1
	daily_DLT	CDC044	COM261	Duplication	34E2220C0B9728F24
	backup_DLT	DC1534	BGD626	Backup	B0E144B5903F3A34
	backup_DLT	DC1532		Backup	5CE114F473736BC3
	backup_DLT	BGD581	BNY612	Backup	DEE1CE64263DF008
	daily_DLT	COM261	BGD624	Backup	2EE0699290907EE
	backup_DLT	AKP387		Backup	A2E17940EFD119E9
	backup_DLT	BNY612	BGD581	Duplication	7DE2176ABCB56F43
	daily_DLT	BGD624	COM261	Duplication	48E1E967CFDFC4C3
	backup_DLT	DC1531		Backup	F4E1BB2F2D2C2E0F
	backup_DLT	BNY548	BNY540	Duplication	64E24222E15F74BE
	backup_DLT	DC1533		Backup	07E115115147F866E
	DLT	BNY607			FBE2373933BD07FF

24 volume displayed, 0 selected, 0 scheduled

You can determine the duplicate of an original at the CLI by using the **ebreport media** command. This command generates a report that lists currently allocated volumes. Refer to "Viewing Reports on Duplications" on page 9-25, and the **ebreport** man page, for more information about this command.

Note: If more than one library unit of a given media type is attached to your EDM, you cannot tell in which LU your duplicate media will reside.

You can also use the **evmstat -c** command to view a list of all catalogued volumes (see the example below). The entries in this list identify original and duplicate volumes, and the original volume for each duplicate. (Refer to the **evmstat** man page for more information about this command.)

*	Mtype	Seq/BC	Name	LU Name	Volume Type	Vol_ID	Original Vol_ID	Generation
C	DLT	BNY574	backup_DLT	offline_0	none	5ED19D47EED28D1C		0
C	DLT	BDE133	backup_DLT	offline_0	duplicate	48D19D5CCFBF3754	CAD19D501B028C8E	0
C	DLT	BDE145	backup_DLT	ex_210_0	original	CAD19D501B028C8E		0

Setting the Maximum Number of Concurrent Duplications

The system is configured at startup to run no more than one duplication at a time. This is based on the assumption that media duplication has at least two drives available for its use. However, you can configure multiple duplications to run concurrently if four or more drives are available. You configure concurrent duplications in the Backup Configuration window of the EDM GUI, or at the command line.

The number of duplications that you can set is based on the total number of drives that media duplication can use. For every pair of available drives, you can increase the number of duplications by 1. For example, with six available drives you can set this number to any value from 1 to 3.

Configuring Concurrent Duplications in the EDM GUI

In the Backup Configuration window, select the Server tab. In the Duplicate Media section, set the Maximum Concurrent Duplications from 1 to 10; the default is 1.

Note: Do not set the maximum number of duplications to a value that is greater than half the number of drives. Otherwise, *significant* performance degradation to the duplications occurs as the duplications must then share the drives among them.

Configuring Concurrent Duplications at the CLI

At the command line, set the number of concurrent duplications by using the **vmdupcfg** command as follows:

```
# vmdupcfg -set -max_dups N
```

where ***N*** is the number of duplications, from 1 to 10.

For example:

```
# vmdupcfg -set -max_dups 4
```

Confirm your change as follows:

```
# vmdupcfg
Media duplication enabled.
No automatic media ejection from the library unit(s).
Tape pad blocks: 1
Max concurrent dups: 4
```

If two separate library units (LUs) are being used in your system, set ***N*** based on the number of drives in the system that are available for duplication. For example, in a system with a two-drive LU and a six-drive LU, you can set ***N*** to 4 if either or both are used for duplication.

Note: If you set the maximum number of duplications at a value that is greater than the number of available drives in the system, a warning message appears. The message contains the total number of drives for a given drive type and the drive type name; for example:

```
WARNING - The server is configured with
10 DLT7000 drives. The max_dups
parameter will be updated to support 6
duplications. This drive configuration
will only support 5 duplications.
```

Verifying the Status of a Duplication

You can verify the status of active, suspended, scheduled, and failed duplications at the command line by using the **vmdupd** command. (Refer to the vmdupd man page for more information.)

- **vmdupd -all** displays all active, suspended, scheduled, and failed duplications, as shown in the example below. For each duplication status, the following information appears:
  - the trail that is being backed up
  - duplication mode (new or append)
  - volume type
  - duplication status (ACTIVE, SUSPENDED, CANCELED, SCHEDULED, or FAILED)
  - original volume ID
  - sequence number
  - duplicate volume ID
  - total number of blocks that are to be duplicated

(Refer to "Viewing Reports on Duplications" on page 9-25 for more information about duplication status.)

```
# vmdupd -all

Media duplication daemon started.
No Active Duplications found
No Suspended Duplications found
No Scheduled Duplications found
Failed Duplications
Trail: backup_DLT  Mode: append  Type: DLT tape  Status: CANCELED
Orig Vol: FBD79BDAD25C1C62 (BDE099)  Seq #: 000005 in TLU: atl_3264_0
Dup  Vol: None
Total Blocks to Copy: 12817
Trail: backup_DLT  Mode: append  Type: DLT tape  Status: CANCELED
Orig Vol: A8D7969E4D438EE3 (BDE137)  Seq #: 000003 in TLU: atl_3264_0
Dup  Vol: None
Total Blocks to Copy: 75
```

If a Duplication Was Scheduled for an Offline Volume

If you move offline a volume that is scheduled for duplication, the duplication still appears in the output when you run **vmdupd -all**. If you want to remove the scheduled duplication, you must run vmdupd -cancel to designate it as Failed, and then run vmdupd -remove to remove the duplication from the Failed queue.

Manually Disabling and Re-enabling Duplication

Though media duplication is enabled automatically, you can manually disable and re-enable media duplication system-wide at the command line. However, disabling duplication is not recommended unless absolutely necessary (for example, if duplication of original volumes is not to be performed for any volumes in the system at any time).

Procedures for disabling and re-enabling duplication are described below.

Note: Any configuration values that you set for duplication (pad blocks, maximum number of duplications) are lost when you disable duplication. You must reset them when you re-enable duplication if you do not wish to use the default values.

Disabling Duplication

You disable duplication by halting the vmdupd daemon that controls media duplication, and then disabling duplication. Any duplications that are in progress are also suspended, but are completed when duplication is enabled and the vmdupd daemon is restarted.

Use the following procedure at the command line to disable media duplication for the entire system:

1. Run **vmdupd -halt** to halt the vmdupd daemon.
2. Verify with the following command that the daemon no longer exists:

   ```
   # ps -aef | grep vmdupd | grep -v grep
   ```

   The **ps** command enables you to view information about active processes. Refer to the **ps** man page for more information about this command and its options.
3. Run **vmdupcfg -reset** to disable duplication.

Re-enabling Duplication

The re-enabling process also involves two steps; you restart the vmdupd daemon and then enable duplication. Any suspended duplications are then completed.

Use the following procedure to re-enable media duplication for the entire system:

1. Run /usr/epoch/bin/vmdupd & to restart the vmdupd daemon.
2. Verify that the daemon started with the command:

   ```
   # ps -aef | grep vmdupd | grep -v grep
   ```

   An example follows:

```
client:> ps -aef | grep vmdupd | grep -v grep
root 2424 1 0 09:53:05 pts/2 1:12 /usr/epoch/bin/vm
```

3. Run **vmdupcfg -set** to enable duplication; remember that using this command resets the duplication parameters back to the default values.
4. Verify the operation by using **vmdupcfg** with no arguments:

```
# vmdupcfg

Media duplication enabled.
No automatic media ejection from the library unit(s).
Tape pad blocks: 1
Max concurrent dups: 1
```

Note: You need to reset any values (tape_pad, max_dups) that were previously set before duplication was disabled.

Pausing, Resuming, Canceling, or Removing a Duplication

You can temporarily pause duplication in progress at the command line (for example, if it is necessary to allow other operations to complete).

Note: Before pausing, verify with the **vmdupd -all** command that duplication is ACTIVE.

Pausing Duplication

To pause duplication, use the command **vmdupd -stop**. After entering **vmdupd -all** again, notice that the state changed from ACTIVE to SUSPENDED. This shuts down the duplication process and volumes are dismounted from drives. In the Library Unit Manager (LM) window, also notice that the application ID of "MD" disappears from the volumes that are being used for duplication.

Note: This command disables scheduling of all duplications system wide.

Resuming Duplication

To continue a paused duplication, use the command **vmdup -cont**. Verify with **vmdupd -all** that the state of duplication changed from SUSPENDED to ACTIVE (allow a few moments for the state to change). In the Library Unit Manager (LM) window, the volumes are remounted. Notice that the application ID of "MD" reappears on the volumes in use.

Canceling Duplication

You can cancel an active, suspended, or scheduled duplication. Canceling the process sets the duplication state to FAILED.

Canceling an active duplication shuts down the process; when the duplication is rescheduled, the entire volume is duplicated.

To cancel a duplication, you use the following command:

```
# vmdup -cancel -volID <orig volID>
```

where *orig volID* is the ID of the corresponding original volume.

The resulting output of the canceled duplication is similar to the following. The value within parentheses is the volume's barcode.

```
# vmdup -cancel -volID 7DD7E5CDDCD690DB

Duplication for volume 7DD7E5CDDCD690DB (AAC009)
canceled
```

If you wish to check on the status of the canceled duplication, you can use the **vmdupd** command. The duplication status should appear as FAILED, as shown:

```
# vmdupd -all
Media duplication daemon started.
No Active Duplications found
No Suspended Duplications found
No Scheduled Duplications found
Failed Duplications
Trail: backup_DLT  Mode: append  Type: DLT tape  Status: FAILED
Orig Vol: 7DD7E5CDDCD690DB (AAC009) Seq #: 000004 in TLU: de_x7
Dup  Vol: None
Total Blocks to Copy: 15445
```

Removing a Failed Duplication from the Queue

You can remove a failed duplication from the queue by using the **vmdup -remove** command at the CLI. Using this command deallocates all duplicate volumes that are associated with the original volume, which makes them available for reuse.

The **vmdupd -remove** command also clears the duplication flags for the original volume, which removes the original volume from the failed duplication queue, and removes the duplication state of Failed from the original volume.

If another backup is scheduled for the original volume, the duplication of that volume is also scheduled.

Use this command as follows:

```
# vmdup -remove -volID original volume ID
```

For example,

```
# vmdup -remove -volID 5DDC2A1510B0A3D0

Volume 5DDC2A1510B0A3D0 removed from the failed
duplication queue.
```

You can then use the **vmdupd -all** command to verify that this operation is successful.

If a Duplication Fails

During media duplication, the duplication is set to FAILED if:

- you reject a volume request for a specific duplicate volume at the beginning of a duplication process
- you reject a queued request for an available duplicate volume (the queued request no longer reappears)
- a request occurs for mounting and relabeling a duplicate volume that is considered "unmountable" (for example, barcode mismatch)
- you cancel a duplication (refer to "Canceling Duplication" on page 9-17)
- a read error occurs on the original volume
- a write error occurs on the duplicate volume

Be sure to reschedule the duplication as soon as possible so that duplication of the specific volume can resume after backup. Otherwise, if the trail supports automatic duplication, subsequent backups to that specific trail do not schedule duplications automatically until the next rotation.

Rescheduling a Failed Duplication

If a duplication failed for some reason, you can reschedule it in the Library Unit Manager (LM) window of the EDM GUI, or at the command line.

Note: Always check the uses count on the duplicate that is still mounted in the drive to ensure that it is less than the maximum uses for the volume. This prevents the volume status from changing to Expired for the mounted volume. (If a mounted volume's status changes to Expired, you must manually remove the volume from the drive.)

Rescheduling a Failed Duplication Through the GUI

In the LM window, click on the Duplications button. The Media Duplication Control window that appears lists all failed duplications. In addition to the Failed status, information for the failed duplication also includes the trail name and sequence and bar code numbers for the original and duplicate volumes. (Refer to "Viewing Reports on Duplications" on page 9-25 for information about rescheduling a failed duplication.)

To reschedule a duplication, select the volume for duplication in the Media Duplication Control window and then click on the Reschedule button. (Holding the Shift key while selecting enables you to select more than one volume at a time.) Information for the selected volume then disappears from the window.

(You can also access the Media Duplication Control window in the EDM Main window by selecting Duplications in the Media pull-down menu, or by selecting a Library Unit in the window, clicking on the right mouse button, and choosing Duplications from the pop-up menu.)

Note: It is important that failed duplications be rescheduled, as no subsequent backups to the original are duplicated until rescheduling occurs and duplication is successful.

Rescheduling a Failed Duplication at the CLI

When rescheduling a failed duplication at the CLI, you can view all failed duplications by using the **vmdupd -all** command as follows:

```
#vmdupd -all

Media duplication daemon started.

No Active Duplications found

No Suspended Duplications found

No Scheduled Duplications found

Failed Duplications
Trail: misk_ms_DLT  Mode: append  Type: DLT tape  Status: FAILE
Orig Vol: E1D86D2C8219CB14 (BDE145)  Seq #: 000018 in TLU: atl_
Dup  Vol: None
Total Blocks to Copy: 400
```

Then enter the following:

```
# vmdup -reschedule -volID volume_ID
```

where *volume_ID* is the original's volume ID. The failed duplicate is deallocated and an available volume is selected.

Following is an example of the resulting output (the barcode, if any, appears in parentheses after the volume ID):

```
# vmdup -reschedule -volID E1D86D2C8219CB14

Volume E1D86D2C8219CB14 (BDE145) rescheduled for
duplication
```

Use the **vmdupd -all** command to verify that the duplication is scheduled:

```
# vmdupd -all
Media duplication daemon started.

No Active Duplications found

No Suspended Duplications found

Scheduled Duplications

Trail: backup_DLT  Mode: append  Type: DLT tape  Status: SCHEDULED
Orig Vol: E1D86D2C8219CB14 (BDE145) Seq #: 000024 in TLU: atl_3264_0
Dup  Vol: None
Total Blocks to Copy: 19629
```

Note: You cannot reschedule a volume that is already scheduled for duplication; otherwise, an error message appears. You must first cancel the duplication by using the command **vmdup -cancel -<*volume_ID*>** and then reschedule.

Rescheduling Duplication of an Offline Original Volume

If you try to reschedule duplication of an offline original volume at the CLI, a message appears that states that the original is offline. For example:

```
# vmdup -reschedule -volID <volume ID>

Cannot schedule duplication for original volume:
<volume ID> (barcode, if any), volume is in library
unit: offline_0
```

You must then inject the offline volume into the library unit before rescheduling it.

Rescheduling Duplication of a Single Volume for Archival Purposes

An occasion may arise in which, for archival purposes, you want to duplicate a volume of a trail that is not normally scheduled for manual or automatic duplication.

Before rescheduling the volume for duplication, first run **ebreport media** to find the volume ID of the volume to duplicate:

```
# ebreport media

Rotations for Template "default", Trail "backup_DLT", Primary Trailset

12/31/1999 18:03:03 Rotation ID:56D874D8.106F916B.00000200.390B91E2, 35 backups

     Media duplication not used

*Orig Vol: 56D874D8106F916B (BDE132), Seq #: 000026 in TLU: at_452_0, media: DL

Orig Vol: 13D874E52C75916D (BDE023), Seq #: 000032 in TLU: at_452_0, media: DLT
```

Next, use the **vmdup** command to reschedule duplication of the selected volume:

```
# vmdup -reschedule -volID <volume ID>
```

Following is an example of the resulting output (the barcode, if any, appears in parentheses after the volume ID):

```
# vmdup -reschedule -volID 56D874D8106F916B

Volume 56D874D8106F916B (BDE132) rescheduled for
duplication
```

Then, by using the **vmdupd -all** command you can verify that the duplication is scheduled.

```
# vmdupd -all
Media duplication daemon started.

No Active Duplications found

No Suspended Duplications found

Scheduled Duplications

Trail: backup_DLT  Mode: append  Type: DLT tape  Status: SCHEDULED
Orig Vol: 56D874D8106F916B (BDE132) Seq #: 000024 in TLU: at_3264_0
Dup  Vol: None
Total Blocks to Copy: 19629
```

Restoring from Backup or Duplicate

When you restore a backup, you can use either an original backup volume or an up-to-date duplicate volume. If the original volume is physically present in a library unit (that is, not offline or offsite), the original is automatically used for the restore.

Note: An up-to-date duplicate implies that no additional backups were appended to the original since this duplicate was completed. Therefore, the duplicate volume is an exact duplicate of the original

If the original volume is offline or offsite but an up-to-date duplicate volume is physically present in a library unit, the duplicate volume is automatically used.

If both the original and up-to-date duplicate are offline or offsite, processing suspends until appropriate media is injected into the library unit. Within the EDM GUI, the Volume Request window appears, which prompts you for the original or current duplicate. You must then load either volume into the library unit so that the restore process can use it.

Note: The duplicate volume is not substituted for the original volume during a restore if the original was modified since the last duplication.

If an Original Volume is Defective

If an original volume in your library unit is defective and a valid, current duplicate is available offline, eject the original volume so that it is no longer used for restoring files or doing additional backups.

Viewing Reports on Duplications

You can use the **ebreport media** and **ebreport duplicate** commands to check the status of duplicate media. The reports that these commands generate are described below.

(Refer to the **ebreport** man page for more information.)

ebreport media Report

The report that **ebreport media** generates reports lists all currently allocated volumes by volume ID and barcode (where applicable), and identifies whether a volume is an original or a duplicate. Also listed is whether duplication is enabled for a rotation of a trail, and if so, how many duplicates were made.

A sample report is shown:

```
# ebreport media

EDM Backup Media Report for server edm on Dec 31 23:07:40 1999
Report options: none

Summary of all media, listed by media rotation groups

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset

12/31/1999 09:46:51 Rotation ID:A1D7F9BD.71812B77.00000200.F206F11B, 104 bac
     Media duplication used on 1 copy

*Orig Vol: A1D7F9BD71812B77 (BDE098), Seq #: 000025 in TLU: at_3264_0, media: DLT ta
 Dup  Vol: 45D81F75C195EEF0 (ASV891), Seq #: 000027 in TLU: at_452_0, media: DLT tap
       Duplication State Active, Empty,  Duplication Date 12/14/1999 13:43:
 Dup  Vol: 40D61EE7477F8BDA (BDE146), Seq #: 000017 in TLU: at_3264_0, media: DLT ta
       Duplication State Done, Successful,  Duplication Date 12/31/1999 13:
```

If the trail supports duplication and no currently allocated duplicate volume is available for the original, "None" appears for duplicate volume information, as shown:

```
12/31/1999 23:57:57 Rotation ID:65D8498A.FD4DC69B.00000200.540819F4, 2 backups
     Media duplication used on 1 copy
*Orig Vol: 65D8498AFD4DC69B (BDE012), Seq #: 000020 in TLU: at_452_0, media: DLT
 Dup  Vol: None

     State Done
```

Note: If you see "None" in a report, you should research the reason for the entry further. This may indicate that a duplication process did not run correctly.

For each rotation of a schedule template, just before the volumes are listed, one of the following appears on the screen:

```
Media duplication used on 1 copy

Media duplication not used
```

Note: When duplication is enabled for a trail, the report always states that duplication is used.

The volume state follows the word "State" and appears as Scheduled, Active, Suspended, Resuming, Failed, Done, or Imported.

The duplication status follows the volume status and appears as Empty or Old (another backup was appended to the original volume but not duplicated), depending on whether the duplicate is valid. (Refer to "Importing a Duplicate Volume" on page 9-29.)

The duplication volume states are listed in on page 9-28.

ebreport duplicate Report

The report that **ebreport duplicate** generates includes the information that the **ebreport media** report contains. In addition, the **ebreport duplicate** report also provides the mode of duplication (append or new), the total number of blocks that were duplicated, the start and stop times of the duplication, the end time of the last duplication, and the duplication expiration date. A sample report is shown:

```
# ebreport duplicate

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset

12/31/1999 09:46:51 Rotation ID:A1D7F9BD.71B12B77.00000200.F206F11E, 104 backups
     Media duplication used on 1 copy
Duplication State: Done, Old, Mode: New
*Orig Vol: A1D7F9BD71812B77 (BDE098) Seq. #: 000025 in TLU: atl_3264_0, media: DLT tape
 Dup  Vol: 40D81EE7477F8BDA (BDE146) Seq. #: 000017 in TLU: atl_3264_0, media: DLT tape
Total Blocks: 349028  Start Time: 12/22/1999 10:54:26  End Time: 12/22/1999 13:06:21
Duration: 001 Hrs. 31 Min., Duplicate Expiration Date: 12/25/1999

12/31/1999 12:57:57 Rotation ID:65D8498A.FD4DC69B.00000200.540819F4, 2 backups
     Media duplication used on 1 copy
```

If the trail supports duplication and no currently allocated duplicate volume is available for the original, "None" appears for duplicate volume information, as shown below.

Note: If you see "None" in a report, you should research the reason for the entry further. This may indicate that a duplication process did not run correctly.

```
Duplication State: Done, Mode: Append

*Orig Vol: 65D8498AFD4DC69B (BDE012) Seq. #: 000020 in TLU: atl_452_0, media: DLT tape
 Dup  Vol: None

Total Blocks: 207
```

Table 9-1 below lists several examples of duplicate volume states.

Table 9-1 Duplicate Volume States

Volume State	Duplication Status	Description
Done	Successful	A valid, up-to-date duplicate volume is complete.
	Empty	Duplication is complete but no data was written to the duplicate volume.
	Old	Another backup was appended to the original volume since this duplicate was completed. Therefore, the duplicate is not a complete duplicate of the original.
Active	Empty	Duplication of the backup is in progress. Backup information was appended to the original and was queued for duplication. During this time the duplicate shows up as Active, Empty, or Active, Old. (Use **vmdupd -all** to verify.)
	Old	Data was appended to the original backup volume but the duplication is in progress.
Failed	Empty	Duplication of this volume failed for some reason, or duplication was intentionally canceled. Check the detail log (/var/adm/epoch/detail) for more information about the failed duplication. Reschedule this duplication.
Suspended		Duplication of the backup volume was suspended.
Resuming		A suspended duplication is restarting or an appended backup is being duplicated and the duplication is starting.
Scheduled		The backup volume is a candidate for duplication.
Imported		The duplicate volume was imported into the EDM and is ready for use.

Importing a Duplicate Volume

If you import a duplicate volume (using **ebimport**), **ebreport media** reports the status as "Not started, empty," which is erroneous. This status is corrected when the duplicate volume is appended to.

Note: The duplicate volume, when imported, is not substituted for the original volume if the original was modified since the last duplication.

Importing a Duplicate Volume before the Original

If you import a duplicate before the original, a volume catalog entry is also created for the original.

If the original is imported:

1. A dialog asks if you want to delete the existing entry with the same volume ID. You should answer Yes.

   The first entry for the original still appears; a second, uncataloged volume also appears in the same drive.

2. Close the Library Unit Manager window.
3. Click on Volumes to reopen it.

   Now only the uncataloged volume is in the drive and the first entry for the original volume is gone.

4. Now import the uncataloged volume again.

Rejecting a Mount Request

When you reject a mount request through the Media Request window, the scheduled duplication is set to Failed to prevent the duplication from being recursive. This Failed status appears in the detail log. (See "Log File Rotation and Archival" on page 15-4.) System monitoring also flags this status. (See *EDM System Monitoring Report* for more information.)

You should reschedule a failed duplication by selecting it in the Duplications window, or by using **vmdup -reschedule** at the command line. (See "Viewing Reports on Duplications" on page 9-25.)

Note: It is important that failed duplications be rescheduled, as no subsequent backups to the original are duplicated until rescheduling occurs and duplication is successful.

Expiring a Duplicate Volume

You can expire a duplicate volume at any time before or at the same time you expire an original volume. (The original volume should be taken offsite rather than the duplicate because the expiration period of an original volume can be longer than the duplicate.)

In the Media tab of the Backup Configuration window, click on the Expiration Options button. In the Media Expirations window that appears, set the expiration period in years, months, or days. Then click on Apply.

Note: If you attempt to set the expiration date of a duplicate after that of its original volume, an error message appears. You must specify a new expiration date before you can continue.

To expire a duplicate volume at the command line, use the **ebexpire** command. This command enables you to expire original and duplicate volumes as well as saveset records and backup catalogs. (Refer to the ebexpire man page for more information.)

To expire a duplicate volume, enter the following:

```
# ebexpire -d -expire -purge
```

The -d option specifies that a duplicate is to be expired.

Viewing Duplicate Expiration Dates

You can use either of two commands at the command line to view duplicate expiration dates:

1. Using the command **ebreport history -expire_times** lists all expire times for media, catalogs, duplicate media, and savesets. An example follows:

**** Work Items for Template dlt_dup_2, Primary Trailset ****

**Item "dup:/home/client_2" for client "xyz"

Time	Lvl	ID	Status	Entries	Cat_Exp	Dup_Exp	SS_Exp	Med_Exp	Rcvr
12/04/99 14:13	9	72306582.345F741C	complete	412	12/05/99	12/05/00	03/05/00	03/05/00	r
12/04/9913:27	9	72306582.345F6924	delta	0	12/05/99	no dup	03/05/00	03/05/00	
12/04/99 11:19	9	72306582.345F4B50	complete	0	12/05/99	01/05/00	03/05/00	03/05/00	
12/04/99 10:59	9	72306582.345F46B8	complete	0	12/05/99	expired	03/05/00	03/05/00	
12/04/99 10:45	9	72306582.345F435B	delta	0	12/05/99	no dup	03/05/00	03/05/00	
12/30/99 12:57	9	72306582.3458CAF6	complete	0	12/31/99	01/05/00	03/30/00	03/05/00	

2. Using the command **ebexpire -check** scans the saveset database and lists the current state of the selected backup resources, including the status of duplications. An example follows:

```
72306582.345F741C: media state is "onsite", expiring on 12/05/99 14:1
72306582.345F741C: saveset state is "active", expiring on 12/05/99 14
72306582.345F6924: catalog state is "delta", expiring on 12/05/99 13:
72306582.345F6924: duplicate state is "exists", expiring on 12/05/99
72306582.345F435B: media state is "onsite", expiring on 12/05/99 10:4
72306582.345F435B: saveset state is "active", expiring on 12/05/99 10
72306582.3458CAF6: catalog state is "delta", expiring on 12/05/99 12:
72306582.3458CAF6: duplicate state is "expired"
```

10 Magnetic Disk Concepts

To keep your system working smoothly, you must monitor disk space and catalogs, and perform some system maintenance. Although EDM Backup software has commands and scripts to control disk space usage, you should also monitor it periodically.

This chapter covers the following topics:

- Expiration of Backups and Catalogs
- Filesystem Cleanup Script
- Magnetic Disk Capacity
- Managing Disk Space

Expiration of Backups and Catalogs

On occasion, you need to expire old backups and backup catalogs. Expiring backups frees up storage media for reuse as well as the disk space that their corresponding backup catalogs use on the server. Expiring additional catalogs of older backups can free up additional needed disk space on the server.

EDM Backup software creates a backup catalog each time it backs up a work item. The backup catalog identifies a backup at the file level by recording the names and attributes of each file in the work item at the time of the backup. It also keeps track of the location of backed up data for each file that was selected for backup. You need the backup catalog when restoring files.

Catalogs are stored online on the server and can grow to be quite large. To maintain sufficient magnetic disk space you must expire the catalogs after a fixed period, perhaps quite earlier than you wish to expire the backups themselves.

You can expire a catalog and still restore data from its corresponding backup if necessary, because unprocessed catalogs are also stored directly on the backup tapes along with the backup data. If you need to access an old backup, you can recreate the catalogs from these raw, unprocessed catalogs on the media by using the **ebimport** command.

For each backup of a work item, the server also creates an online saveset as well as the backup catalog. The *saveset* record contains information about an entire backup, for example, its start time, the media trail that the backup program used to write the backup data, and the expiration periods. The saveset records do not occupy much space.

Choosing Expiration Periods

The Media Expirations pop-up window enables you to set fixed periods for expiration of backup data, catalogs, and the saveset record. You reach this pop-up window through the Media tab of the Backup Configuration window.

You should change the timing for expiring catalogs, backups, and savesets according to these rules:

- Backup period must be greater than twice the rotation period because incremental backups on one tape in a media set (trailset) assume access to the previous incrementals and most recent full backups on previous tapes in that media set.

  Note: It is important for all of these expiration periods to be greater than twice the rotation period. The reason for this is that you need a full backup and subsequent delta catalog files to reconstruct your system for disaster recovery. It is especially important if you automate deletion of expired backups in crontab.

- Catalog period must be greater than twice the rotation period, but it can be less than the backup period.
- Saveset period must equal the backup period unless you choose never to expire the backup period. In no case can it be less than twice the rotation period.

You can change the expiration periods at any time. Any changes to the expiration periods take effect the next time **ebbackup** runs, either from an entry in root's crontab file or manually by typing the command **ebbackup** from the command line.

```
# /usr/epoch/EB/bin/ebbackup
```

Running Expiration

After a backup is eligible for expiration, you can delete it from your system, either automatically by running the command **ebexpire -purge** in root's crontab, or manually from the command line. Use the **-purge** option with the **ebexpire** command when you want to delete expired catalogs.

```
# /usr/epoch/EB/bin/ebexpire -purge
```

The **ebexpire -purge** command identifies backup data, catalogs, and saveset records that are eligible for expiration because they exceeded their duration period as set in the backup configuration. (Refer to the **ebexpire** man page for more information about this command.)

Filesystem Cleanup Script

Try to keep the number of old and unneeded files to a minimum on your system to conserve space for the backup catalogs. The cleanup script **epcleanup** simplifies the cleanup of your filesystem by removing unneeded or old files from tmp, crash, and adm subdirectories.

The system is automatically configured to execute the **epcleanup** script from **cron** with the default settings. The line in root's crontab file that specifies the **epcleanup** script is:

```
30 8 * * * /usr/epoch/lib/epcleanup > /dev/null
2>&1
```

If you want to override any of the defaults, specify the option name and the new value in the crontab file. For example, if you want to change the number of days to expire unmodified log files from 14 days (the default) to 10 days you enter:

```
30 8 * * * /usr/epoch/lib/epcleanup -log 10 >
/dev/null 2>&1
```

Magnetic Disk Capacity

You must know your system's catalog capacity to be able to configure your site for optimal performance.

EDM Backup software consumes magnetic disk space based on the total number of files to back up. In addition, disk space consumption is affected by the rotation period (the number of days between level 0 backups), the expiration period for catalogs, the percentage of files that change on a daily basis, and the average catalog size per file.

The calculations in this section assume that:

- the system is used as a backup server only.
- five percent of the files are modified each day; this affects incremental backups (see "Calculating Actual Daily File Changes" on page 10-7).
- catalog size averages 200 bytes per file.
- catalog requirements for database backup are small.

Rotation and Keep Catalog Periods

Keep in mind that the *minimum* keep catalog period must be at least twice that of the rotation period to ensure that the catalog is relieved of its dependencies. The system defaults are a rotation period of 14 days and a keep catalog period of one month.

Table 10-1

Minimum Keep Catalog Period (Days)

Rotation Period		
7 Days	**14 Days**	**28 Days**
14	28	56

Table 10-2 shows the *maximum* number of days that you can use for the keep catalog period for a 25.2 GB catalog subsystems as you vary the number of files that you back up and the rotation period.

Pick the table that matches the catalog disk size that you have. Then select the number of files to back up. You can then refine the suitable rotation period and keep catalog policy.

As long as you are well within the maximum limit of files, you are free to adjust rotation and keep catalog periods up and down. As you approach the upper limit of files, you run into some constraints, as noted in Table 10-2.

Table 10-2

25.2 GB Catalog Subsystem: Max. Keep Catalog Period (Days)

Number of Files (Millions)	Rotation Period		
	7 Days	14 Days	28 Days
1	470	745	1045
2	235	370	520
3	155	245	350
4	115	185	260
5	95	150	210
6	80	125	175
7	65	105	150
8	60	90	130
9	50	80	115

Table 10-2

25.2 GB Catalog Subsystem: Max. Keep Catalog Period (Days)

	Rotation Period		
10	45	75	105
12	40	60	85
14	35	55	75
16	30	50	70
19	25	40	56
23	20	30	< 56[1]
31	15	< 28[1]	< 56[1]

1. Not allowable because the maximum is less than the minimum of 2x the rotation period.

Calculating Actual Daily File Changes

Table 10-1 and Table 10-2 above assume that five percent of the files are modified each day. By looking at a backup history report, you can calculate the actual number of files that change each day at your site.

This affects the incremental backups. If the actual percentage is less, additional files can be backed up. If the actual percentage is more, fewer files can be backed up.

You must wait until EDM Backup has been running for one rotation period before you can generate enough information to determine the number of daily file changes.

The catalogs for incremental backups are consolidated to reduce storage space on the server. All but the most recent incremental backup catalog are turned into *deltas*, which list only backup files that differ from those that are listed in subsequent catalogs.

Note: If you custom schedule an explicit incremental backup (level 1-8), by default, the catalogs for these backups are *not* consolidated. You can change the *backup catalog delta level* field in the Backup Configuration window from 9 to the lowest integer level for which you want consolidated catalogs.

You must review the number of entries in a *delta catalog* to determine your daily changed file rate. A delta catalog contains only the information that differs from the subsequent catalog files. The number of entries in a delta catalog represent the number of files that were changed during the time between the backup that delta represents and the subsequent backup. Thus, the size of a delta catalog represents a measure of how many files were changed during a backup.

Run a backup report (**ebreport backup**) to display the number of files or directories (entries) in each backup catalog. Figure 10-1 illustrates a report that was produced with the **ebreport backup** command, and specified with the **-since** option and a date. The report shows the type of backup catalog that was created on this date, and the number of file entries.

Figure 10-1 **ebreport backup -since Report**

```
EDM Backup Backup Report for server adam at Oct 24 13:44:05 1998
Report options: -since 9/15/98


Template name,  Primary/Alternate: Trailset name
================================================
default,        Primary: primary

Work item name                    Level   Start time Time used BackupFiles\bad Size
Catalog
-------------------------------------------------------------------------------
------------------
adam:/                            9   9/23/98 19:33:03   0:12:22 Completed2695\0214
MB Unsorted
adam:/                            9   9/22/98 19:42:24   0:13:42 Completed926\0115.
MB Complete
adam:/                            9   9/21/98 18:14:37   0:11:24 Completed234\00.0 I
Delta
adam:/                            0   9/21/98 15:44:04   0:43:48 Completed26720\0
```

To determine the number of files that are backed up each day, run a backup history report for a complete rotation period, not including the last backup run. For example:

```
emc# ebreport history -since 9/6/98 -until 9/19/98
```

This command produces a report that covers a 14-day rotation period (refer to the **ebreport** man pages for more information). The command output provides a line for each backup of each work item, which shows the catalog type and the number of entries in the backup catalog.

For example, you can view a report that contains lines that are similar to those in Figure 10-2.

Figure 10-2 **ebreport history -since -until Report**

```
Backup History Report for server atlas1 on Nov 20 16:06:15 1997

**** Work Items for Template cad-all, Primary Trailset ****

**Item "cad5-all"

TimeLvlIDStatusEntriesExpire

1/ 7/97 20:010C005531B.296A4F13complete317661/ 6/98

1/ 6/97 20:019C005531B.296A32E9delta39431/ 5/98
```

Backup Status (delta or complete)

Managing Disk Space

To keep your EDM system working smoothly, you must monitor your magnetic disk space. This section describes several ways to do this.

Distributing Catalogs

After you run EDM Backup software for awhile, you must split the backup catalogs proportionately among the available space on each disk. EDM Backup software creates catalogs for each backup to track the file data in the backup. Initially, EDM Backup software places all backup catalogs in a single partition under the directory (or symlink) /usr/epoch/EB/catalogs.

The installation procedure creates the directory /usr/epoch/EB/catalogs. In the catalogs directory each work item has one subdirectory – named for that work item. EDM Backup places all catalogs produced for that work item in that subdirectory, even if the work item is backed up through multiple templates and trailsets.

Note: You must keep the catalogs on disks that are local to the backup server, not on an NFS-mounted filesystem.

To split the backup catalogs among the disk partitions, use the following procedure:

1. Run **ebbackup** until every work item had at least one successful backup through each of the configured schedule templates and trailsets (media sets).

   Use the **ebreport history** command to determine that all work items had a successful backup. For information on using this command, see the **ebreport** man page.

2. Divide the work items into one group per disk partition so that the total number of catalog entries in each group is in proportion to the size of the partitions. You can determine catalog entries per group from the **ebreport history** report output.

   To determine the number of catalog entries in a work item, run the **ebreport history** command. For each work item, add the number of entries from the most recent complete catalog for every template and trailset. This sum is the size of the backup catalogs that EDM Backup places in the work item's directory during a rotation period.

   For example, consider a site that has 50 work items of the same size (CAD1 through CAD50) and three available disk partitions for backup catalogs. Two of the disk partitions have 900 MB of free disk space and the third disk partition has 400 MB. To divide the total catalog entries proportionately to the ratio of available disk space, the site places 21 work items on each of the 900 MB partitions, and eight work items on the 400 MB partition. Figure 10-3 below illustrates this distribution.

Figure 10-3

Work Item Distribution

3. Move the individual work item entries for each group of work items onto the target disk partition.

   Note: Do not move a work item directory while backups are running or while catalogs are being processed.

   For example, assume that /usr/epoch/EB/catalogs is a symbolic link to /home/EB/catalogs and that all catalogs were initially on the /home partition in that directory.

   To move the work items cad10 through cad20 from /home/EB/catalogs on the disk partition /home to the disk partition /data1, move the individual subdirectories for each of these work items from /home/EB/catalogs to /data1/EB/catalogs.

   For example, using the Bourne shell, type:

```
edm# mkdir /data1/EB
edm# mkdir /data1/EB/catalogs
edm# cd /usr/epoch/EB/catalogs
edm# sh
# for wi in cad1? cad20; do
> tar cf - $wi | (cd /data1/EB/catalogs; tar xf -)
> rm -fr $wi
> ln -s /data1/EB/catalogs/$wi .
> exit
```

4. Monitor the storage usage of the partitions. If one partition fills up, move the work item subdirectories to the other partitions to keep the proportions balanced.

Do not move a work item directory while EDM Backup is running or while catalogs are being processed. Always make sure that /usr/epoch/EB/catalogs has a symbolic link to the catalogs in the work item directory. Also, you must keep catalogs on disks that are local to the server. For example, catalogs must be on a local filesystem and not on an NFS mounted filesystem.

Reclaiming Magnetic Disk Space

If you do not have HSM, when your magnetic disks fill to capacity, your backups cannot complete successfully. You know that you exceeded magnetic disk capacity on your system when:

- you see numerous messages similar to:
  `edm: /home: file system full`
- you see numerous messages that backups have failed.
- the **df** command shows that one or more of the local filesystems used over 100 percent of its magnetic disk space.

Manually Expire Unneeded Catalogs

To create space on your disks and to get backups running again, use the following procedure:

1. Run the **ebcatclean** command to expire unneeded catalog related files.
   - If running this command gives you enough magnetic disk space to operate, you can stop at this point.
   - If running this command does not provide enough disk space to operate, continue with the following steps to provide additional space.
2. Run **ebreport history** to see the state of all backup catalogs. You want to ensure a complete catalog to support the delta catalogs you save.

3. Choose a date for expiring backup catalogs. Do not expire up to within two times the rotation period.

   For example, if the rotation period for the specified schedule template is 2 weeks, make sure that the **-until** date (in **ebexpire in the next step**) is at least 4 weeks plus one day ago.

4. Run **ebexpire** with the **-c** switch (to expire the backup catalogs without expiring their backups), and the options **-purge**, **-template**, **-since**, and **-until** for the specified schedule template during the specified dates.

   CAUTION: **If you do not specify the -c option, you will also expire the backup data and the saveset record, which means that the backup is completely unrecoverable.**

If you expire the backup catalogs without expiring their backup, you will still be able to restore your data. Unprocessed catalogs are stored on the backup tapes. You can bring the catalogs online with **ebimport** and then restore the data.

Other Options

The default configuration places a relational database on the /usr filesystem. If that filesystem fills up, and you cannot identify any files to delete or relocate, you can try one of the following:

- Rebooting to free any space that may have been used by dead processes to hold open deleted files
- Expiring backup media and relocating the database to a filesystem other than /usr
- Increasing the size of the /usr filesystem

If necessary, contact customer service for assistance.

Changing the Automatic Cleanup Script Defaults

Try to keep the number of old and unneeded files to a minimum on your system to conserve space for the backup catalogs. The cleanup script **epcleanup** simplifies the cleanup of your filesystem by removing unneeded or old files from /usr/epoch/tmp, /usr/epoch/adm, and /usr/epoch/etc.

The system is automatically configured to execute the **epcleanup** script from **cron** with the default settings. The line in root's crontab file that specifies the **epcleanup** script is:

```
30 8 * * * /usr/epoch/lib/epcleanup > /dev/null 2>&1
```

If you want to override any of the defaults, specify the option name and the new value in the crontab file. For example, if you want to change the number of days to expire unmodified log files from 14 days (the default) to 10 days you enter:

```
30 8 * * * /usr/epoch/lib/epcleanup -log 10 > /dev/null 2>&1
```

Part II Hierarchical Storage Management (HSM)

11 Basic HSM Concepts

EDM Backup with HSM Option and EDM Migration client software extend filesystem space by migrating file data on magnetic disks out to optical disks, magnetic tapes, or even other magnetic disks, which creates a much larger *virtual* filesystem. All files, even those that are staged out, appear to the user to be resident on the local magnetic disk.

EDM Backup with HSM Option software provides HSM for the local server. EDM Migration client software extends HSM support to network clients. To enable network migration, you must configure HSM on both the EDM and the network clients.

This chapter introduces some basic migration terms and concepts that provide you with background information for performing tasks.

These concepts include:

- When Files Stage In and Out
- Filesystem Configuration and Maintenance
- File Control Properties
- Compaction of Staging Media
- Compacting Baseline Media

- Migration Reports
- Baseline Backup
- Restaging Data
- Backup Completeness

When Files Stage In and Out

There is a limitation of a maximum of 2GB minus 1KB for the size of files to be staged out or staged in.

A file is staged out:

- as part of nightly system maintenance. This is called *periodic* stage out because the staging occurs on a schedule (see Figure 11-1). You set up nightly staging runs to reduce disk utilization to a predetermined level, called the low watermark (LWM). You can set up periodic staging runs through root's crontab file.
- during daily system operation when magnetic disk space usage reaches a predetermined level called the high watermark (HWM), or when the filesystem runs out of disk space. This is called *event-driven* or *demand* stage out because it is triggered by a growth in disk usage.
- in response to a user's request. Users can explicitly stage out files with the **emstage** command and stage in files with the **embsi** command.

HSM stages a file back in when:

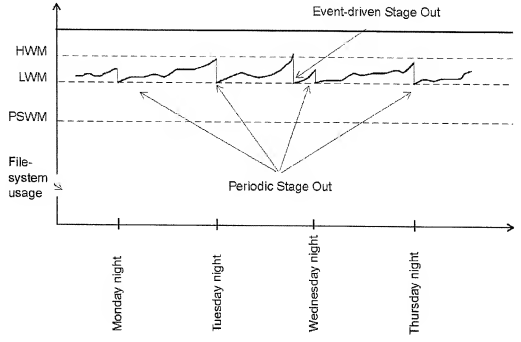
- the staged-out file is read, or
- the staged-out file is modified

When a user reads a staged-out file, migration locates the file and stages it in. At this point, the file is considered *prestaged* – that is, the file resides on magnetic disk *and* on the staging media. Migration may later free the local magnetic storage for this file, without having to repeat the stage out process.

When a user modifies a staged-out file, migration stages in the file and then deletes the staged-out version. If migration later needs to reclaim magnetic storage, it must stage out this file again.

Figure 11-1 shows event-driven and periodic staging over a four-day period.

Figure 11-1 **Staging Out File**



Filesystem Configuration and Maintenance

Configuring HSM is a matter of determining how you can best tune your filesystems so that the most active files are readily accessible. This section discusses several basic concepts that are pertinent to configuration, including:

- staging templates and staging trails

- bitfiles and client stores
- watermarks
- periodic staging and filesystem delay

Staging Templates and Staging Trails

A staging template defines default values for the filesystems that stage to a particular *trail* of optical disks or tapes, or in the case of network migration, a particular *client store* on the server. These default values include:

- template name
- trail type (EO, DLT, network, etc.)
- volume availability (restricted or unrestricted), which specifies how volumes that belong to this template can be reused after they are compacted. *Restricted* volumes can be reallocated only to the same template.
- self-describing media enable/disable

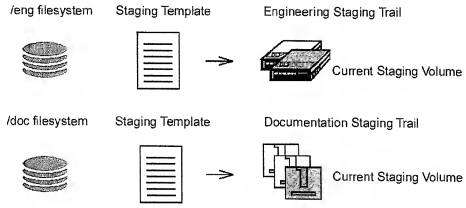
A staging template can also define configuration values such as watermarks and a delay factor. Staging templates exist for filesystems on the migration server and for filesystems on network clients.

When files on the migration server are staged out, they are written to *a staging trail*. Initially, a staging trail consists of a single side of an optical disk or a magnetic tape. Over time, a staging trail can grow to include several optical disks or magnetic tapes. The piece of media that is currently being staged to is called the staging trail's *current staging volume* (see Figure 11-2). Staging trails usually share the same name as the staging template.

When the current volume is full, the Media Requests window tells you to add a new, blank volume to the staging trail. Eventually, this volume fills up as well, and the process is repeated. Over time, this process builds up a trail of staged-out

files for all of the filesystems that are assigned to a template. (Refer to the section on Labeling Volumes for information on allocating a volume to the staging trail.)

Figure 11-2 **Staging Templates and Associated Filesystems**



Deciding How Many Staging Templates to Create

Depending on your site's usage patterns and needs, you can assign all filesystems to a single staging template or assign certain filesystems to their own staging templates. The simplest way to set up staging is to group your filesystems into the fewest number of staging templates as possible. This has several advantages:

- It directs all of your files to a limited number of staging volumes. This reduces the number of staging volumes that need to be moved in and out of library units. When a user requests a staged-out file, there is a greater likelihood that the file resides on the volume that is already in the drive.
- It reduces competition for staging devices. If multiple filesystems begin to stage out at the same time, they can compete with each other for access to staging devices,

leading to a phenomenon called *thrashing*, where the system must repeatedly swap media in and out of drives. Sharing a template prevents this from happening.

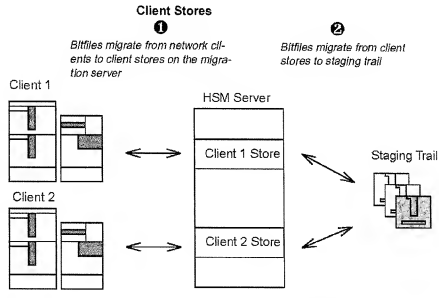
However, there are cases when it is advantageous, even necessary, to create multiple staging templates:

- When you stage files of widely differing sizes. You can reduce the number of mount faults for the smaller files if you keep very large files on a separate staging trail.
- When your site has different access patterns (for example, archival vs. working data). Archival data should be staged separately from working data.
- When you want filesystems that are separate on magnetic disks to also be separate on staging media.
- When you charge groups for the costs of storage media and/or storage space. To keep track of costs, you can create staging templates for each business group and charge them separately.
- When your site expects to expand to multiple servers and move some of the data to a new server. In this case, a separate staging template for each anticipated server simplifies moving the data.

Bitfiles and Client Stores

EDM Migration client software is responsible for the movement of files between network migration clients and the migration server. EDM Migration automatically maintains the relationship between local storage on the client system and server storage, and keeps track of all file data independent of its location.

Figure 11-3



What EDM Migration client software actually stages is a *bitfile* – an uninterpreted bit array. A single bitfile holds the contents of a single client file. Bitfiles are staged to the migration server where they are kept in permanent administrative groupings called *client stores*. Each client system owns one or more client stores on the server. Other clients may be permitted to read these bitfiles, but only the owner client can create or delete the bitfiles.

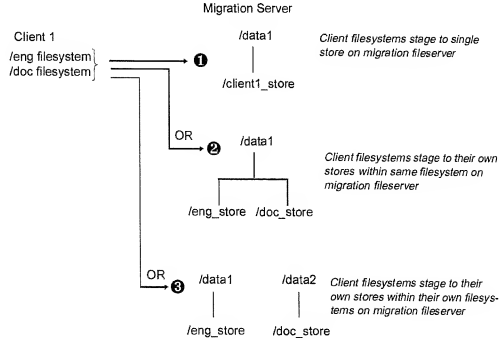
Every client store is associated with a *store ID*, which uniquely identifies it on the network.

Deciding How Many Client Stores to Create

A client store can reside within any migration-enabled filesystem on the migration server. Client stores can all be grouped into a single filesystem on the migration server or spread out among several filesystems. If a network client contains several stageable filesystems, you can stage all of the filesystems to a single store on the migration server (Figure 11-4, example 1); you can stage each filesystem to its

own store within the same filesystem on the migration server (Figure 11-4, example 2); or you can stage each filesystem to its own store within its own filesystem on the migration server (Figure 11-4, example 3).

Figure 11-4 **Client Store Configurations**



As a general rule, the simplest configuration appears in Figure 11-4, example 1, where all of a client's filesystems stage to a single store on the server. If you have three network clients, for example, you would only need three client stores, one for each client. All of the stores would reside within the same filesystem on the migration server.

However, there are cases where it would be advantageous, or even necessary, to distribute the client stores across filesystems. In fact, most of the reasons for creating multiple staging templates (see "Deciding How Many Staging Templates to Create" on page 11-5) are also true for client stores. In addition:

- If a client filesystem consists of a large number of small files, it should stage to a store in its own filesystem. Otherwise, it could cause a filesystem on the server to grow beyond the limit of one million files.
- If you expect there may be a future need to move a client filesystem to another client, you should have that filesystem stage to its own store. This simplifies the move because two clients cannot stage to the same store.

Preventing Redundant Backup of EDM Migration Client Data

If migration client stores are being backed up from the server, there is no need to back up the corresponding data on the client itself. Use the Backup/HSM tag to prevent migration client data from being backed up redundantly.

The Backup/HSM tag is used to update the backupdates file (/usr/epoch/etc/backupdates). When a backup begins for a server work item that has a Backup/HSM tag, a time stamp is entered in the backupdates file. Before a migration client is backed up, the backupdates file is checked. If a file exists on the client that is newer than the client store's work item date listed in backupdates, the file is backed up.

From the EDM configuration interface, you set the tag's value in two places: Backup configuration and HSM configuration.

When you define the work item that backs up the client store filesystem on the EDM (Backup Configuration window, Work Item tab, Work Item options, HSM Options), enter an identifier in the Backup/HSM Tag field. EMC recommends that you use the work item name for the tag. The name must be unique across all work items that back up the EDM's files.

In the HSM configuration window, Client tab, select the tag that you entered in backup configuration for the work item that backs up the store.

If you set the tag from the command line (the emsmks command), note that the tag must match exactly the backup/HSM tag name specified in the definition of the work item.

Full Filesystems

When a magnetic disk becomes 100% full, the system logs a "filesystem full" message via **syslogd**. By default the message is displayed on the console and recorded in the system log file. Being full should be a transient condition for stageable filesystems. Because files are staged out when usage reaches the high watermark, stageable filesystems should rarely fill up.

Watermarks

Migration keeps disk space utilization between user-configurable usage levels called *watermarks*. Watermarks are thresholds that trigger a migration response. Watermarks are expressed as percentages of a filesystem's total disk space, minus the space that some systems withhold from ordinary users. The three watermarks are:

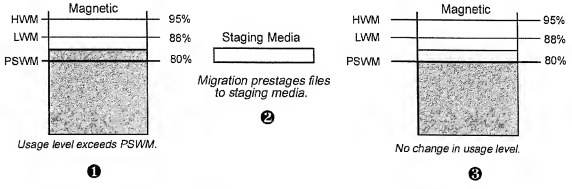
- High Watermark (HWM)
- Low Watermark (LWM)
- Prestage Watermark (PSWM)

To understand how watermarks regulate file migration consider the following example. Assume you have recently installed your migration server and your filesystems still have significant amounts of free space. When users add enough files to cause the usage level to exceed the PSWM, the nightly periodic staging run (set up automatically) will stage files out to the staging media, without freeing any space from the magnetic disk. This is called *prestaging* (see Figure 11-5). Because the

prestaged files still reside on magnetic disk, users can access them quickly. If filesystem usage rises dramatically, migration can simply remove the magnetic images of these prestaged files.

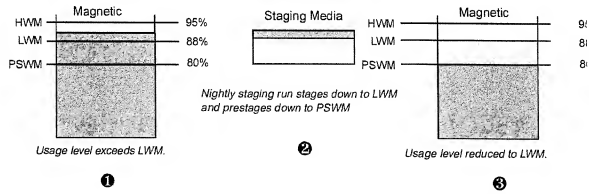
Figure 11-5

Filesystem Exceeds PSWM



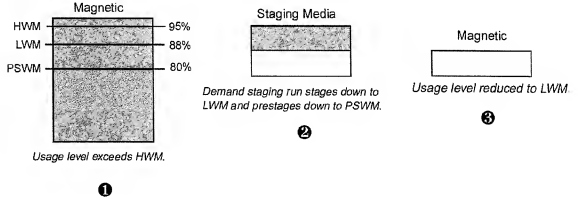
When your users add enough files to cause the usage level to exceed the LWM (88% full), the nightly staging run stages out enough files to bring usage down to the LWM (see Figure 11-6). To do this, migration actually moves files from magnetic disk to secondary storage. In addition, migration prestages files down to the PSWM.

Figure 11-6 **Filesystem Exceeds LWM**



When your users add enough files to cause the usage level to exceed the HWM (95% full), migration triggers a demand staging run, that is, migration immediately stages out enough files to bring usage down to the LWM (see Figure 11-7). Again, migration actually moves files from magnetic disk to secondary storage and prestages files down to the PSWM.

Figure 11-7 **Filesystem Exceeds HWM**



Generally speaking, once you have enough files to fill your magnetic disks, your goal is to keep filesystems in the *green zone*, that is, between the low and high watermarks. A key decision, then, is how large to make the green zone. The optimal size of the green zone depends on your filesystem's usage patterns.

Table 11-1 shows the watermarks for a 1000 MB filesystem. On this filesystem, stage-outs begin when disk utilization reaches 95% capacity, or 950 MB; stage-outs stop when disk utilization falls to 88% capacity, or 880 MB; and migration prestages another 80 MB until disk utilization falls to 80% capacity, or 800MB.

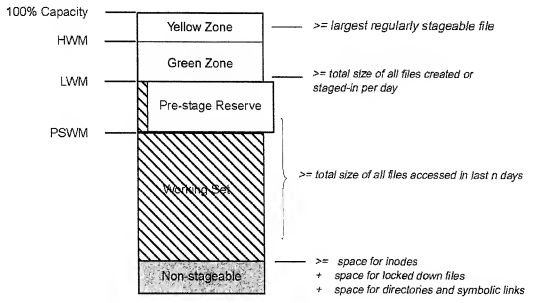
Table 11-1 **Watermarks for a 1000 MB Filesystem**

Watermark	% Capacity	# of MB
HWM	95%	950 MB
LWM	88%	880 MB
PSWM	80%	800 MB

Disk Utilization Zones

The watermarks divide the filesystem into disk utilization zones (see Figure 11-8). The space between 100% capacity and the HWM is the yellow zone, the space between the HWM and the LWM is the green zone, the space between the LWM and the PSWM is the prestage reserve, and the space between the LWM and the non-stageable data is the working set.

Figure 11-8 **Utilization Zones**



The *yellow zone* is reserved for processes to use while migration brings filesystem usage back down to the LWM. It represents the area between 100% capacity and the HWM. Note that if you make your yellow zone slightly larger than the largest regularly stageable file in your filesystem, the system can use this space to stage in or create most any file immediately, while migration then frees additional space by staging out other files, usually from the prestage reserve.

The *green zone* represents the area between the HWM and the LWM, and is the normal zone of operation. The green zone should be large enough to hold the average number of new disk blocks added in a day, including both new files and previously inactive (staged-out) files that are likely to be accessed (staged-in). It should be large enough to make event-driven staging infrequent. Making the green zone larger or smaller is the most common change to the default configuration.

The *prestage reserve* is used for files that have been staged out, but also remain on the system's magnetic space. This magnetic space can be released quickly if disk utilization crosses the HWM. To allow filesystem usage to return to the LWM during a demand-staging event, the prestage reserve is typically the same size, or slightly larger, than the green zone.

The *working set* represents the files that are accessed in a given period of time. For general applications the working set should be in the 7-30 day range. The magnetic disk space holding the working set is actually a combination of the prestaged reserve space (since these files, although staged, are still magnetic resident) and the amount of space available to stageable files below the PSWM. This area needs to be large enough to contain all files accessed in the last n days, where n is the number of days worth of recently accessed files that you want to fit within the working set.

Non-stageable files and other disk structures also consume space; these include space for all directories and symbolic links or for any other files that cannot be staged (for example, swap files).

Sample Watermarks

Consider using one of the sets of watermarks listed in Table 11-2 for your configuration.

The *Archive* settings are designed for filesystems whose files are written once and rarely, if ever, read. The filesystem's data is typically staged-out and rarely, if ever, staged back in. This is the case if large amounts of data are gathered every day and quickly "archived" off of the magnetic disk.

The *Cached* settings are designed for filesystems in which reads outnumber writes, and a relatively predictable set of files are read. This setting takes advantage of migration's ability to keep the most recently-accessed files on magnetic disk, thus ensuring optimal performance.

The *Random* settings are also intended for situations where reads outnumber writes, but where the access pattern is random and least-recently-used caching is ineffective. This would be the case, for example, in a government records office, where several files must be read in from staging media in order to analyze a new file. When the analysis is completed, there is no need to keep the files on magnetic disk, because the files are not accessed again for an undetermined period of time. You can select this setting for filesystems that match this random data access pattern.

You can use these settings as a guide for configuring your filesystems.

Table 11-2 shows the sample watermark settings.

Table 11-2 **Sample Watermark Settings**

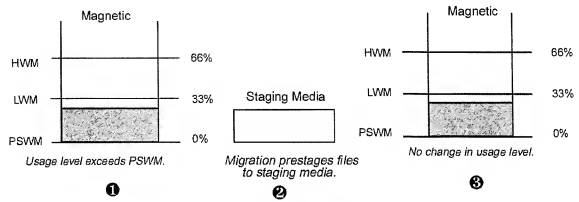
Watermark	HWM	LWM	PSWM
Cached	95%	88%	80%
Archive	66%	33%	0%
Random	68%	34%	17%

The watermarks for Cached filesystems are calculated to maximize the magnetic cache and minimize stale space on staging media.

For archive filesystems, the sample watermarks divide magnetic disk space into three equal zones by setting the high, low, and pre-stage watermarks at 66%, 33%, and 0%, respectively. Any stageable file that is moved onto the disk is prestaged during the next periodic or demand staging run.

Figure 11-9

PSWM Set at 0%



If an archive filesystem contains some temporary files that remains on magnetic disk, you should increase the prestage watermark to take this into account. For example, if a 100 MB archive filesystem is expected to have 20 MB of temporary files, you can set the PSWM at 20% and divide the remaining 80% into equal zones of 26.6%. Thus, the LWM is set at about 47% and the HWM at 74%.

Table 11-3

Watermarks for Archive Filesystems

Watermark	HSM	LWM	PSWM
Default	66%	33	0%
With 20% Temp Files	74%	47%	20%

For filesystems where the random access pattern predominates, the tuning is somewhat similar. The major difference is that when the magnetic disk gets full, it is filled with prestaged files that have been just staged in, whereas archive filesystems are filled with new files. The PSWM is set at 17%. If you expect to fill more than 17% of the filesystem with temporary files, raise

the PSWM and create equal-sized yellow and green zones and a prestage zone of about half the size of the green zone. For example, if a 100 MB Random filesystem is expected to have 25 MB of temporary files, you can set the PSWM at 25%, the LWM at 40%, and the HWM at 70%.

Table 11-4 **Watermarks for Random Filesystems**

Watermark	HWM	LWM	PSWM
Default	68%	34%	17%
With 25% Temp Files	70%	40%	25%

Configuration Issues

Some issues that you need to consider when setting up HSM are the number of files in a filesystem, the type of media you plan to use, and whether or not to enable self-describing media.

Filesystem Limits

Although EMC's HSM products provide virtually unlimited disk space, all filesystems, even stageable ones, are limited by the number of files, directories, symbolic links, and devices they

can contain. For the reasons listed in Table 11-5, EMC recommends that you restrict filesystems to no more than one million files.

Table 11-5 Reasons to Set Filesystem Limits

Reason	Explanation
To ensure a filesystem backup within a single session	Because filesystems provide natural boundaries for backups, having less than one million files on a filesystem allows the entire filesystem to be backed up in one run with no more than 4 work items.
To ensure efficient candidate list creation	Migration selects candidates to stage out by searching all the files (free and used). Although this searching is normally done during off-hours, it also occurs during demand staging runs. The more files in the filesystem, the longer it takes to create the candidate list.
To ensure efficient use of magnetic disk space	Most Unix-based systems maintain an *inode* (also referred to as a *file serial number*) for each file, directory, symbolic link, and device. Each inode consumes a certain number of bytes, depending on the particular operating system.

If you expect a filesystem to consist mostly of many smaller files, or to contain many links, you may find that the filesystem is in danger of running out of inodes. To prevent this from happening, you can decrease the inode density, that is, the number of bytes per inode, at the time of filesystem creation.

Stage-to-Tape Considerations

HSM supports staging to EO, WORM and DLT. In terms of durability and performance, optical disks are the best choice. Although tapes are the most cost-effective media, they are not recommended for applications that require frequent staging-in of data.

In general, staging to tape is only applicable in the following situations:

- In archival environments (where there are very infrequent stage-ins)

- For restaging infrequently-used data to a secondary staging device
- For baseline backups (which use migration technology)

The limitations of tape are:

- Tapes are less durable than optical disks. Whereas DLT tapes are good for tens of thousands to hundreds of thousands of passes, optical disks have no pass count limit.
- DLT tapes have an archive life of around 30 years, but optical disks have an archive life of 25 to 100 years, depending on the manufacturer.
- Tapes are slower. Whereas stage-ins from optical library units usually take less than 20 seconds, stage-ins from DLT tape can take several minutes on a relatively idle library unit, and much longer on a system with high stage-in activity.

Important variable settings for stage-to-tape are described in "Tuning for Staging to Tape" on page 13-3.

Refer to Table 11-6 for a description of the tradeoffs in using stage-to-tape.

Table 11-6 Stage to Tape with Tape Library Units

	Primary staging device 1 drive	Primary staging device 2 or more drives	Secondary staging device 1 or more drives	Baseline backup 1 or more drives	Migrate backup catalogs to tape[1] 1 or more drives
EMC Policy	Disqualified	Conditional	Conditional	Supported	Conditional
Performance	Only capable of servicing ~15 stage-in requests per hour.	Two drives capable of servicing ~30 stage-in requests per hour. Four drives capable of servicing ~60 stage-in requests per hour.	Good, since optical, not tape, will service most user stage-in requests.	Good	Insignificant, since only catalogs are staged.
Media Wear	Significant problem unless used in real archive applications.	Significant problem unless used in archive applications.	Not an issue if data moved to the TLU is accessed infrequently.	Insignificant. Should create low tape access schedule.	Insignificant, since access rate is low.
Deadlock Potential	Significant	Significant	No deadlock cases	No deadlock cases	Reduced, since staging of catalogs is low frequency activity. Can be minimized by careful scheduling of backup and HSM applications.

1. To eliminate thrashing, at least two drives are required, one drive to read in staged data and one drive to write backups to tape.

Self-Describing Media

With self-describing media enabled on the migration server, migration stores the full pathnames of migrated files on the staging media. This allows the media to be moved from a migration server to another server. With self-describing media enabled, however, migration will require more time to stage files.

You can enable or disable self-describing media with the **emstconf** command.

Periodic Staging and Filesystem Delay

As part of your nightly maintenance, you should schedule periodic staging runs of your filesystems to bring disk utilization down to the LWM. Periodic staging runs are set up through root's crontab file.

If you're staging out files from more than one filesystem, you should stagger migration to minimize loads on your system. The actual time that staging begins for each filesystem depends on the filesystem's *delay* parameter, which you can set with the **emfsconf** command. The filesystem delay parameter specifies the number of minutes to wait after the nightly staging run is scheduled to start before beginning stage-outs for a given filesystem.

In setting the delay, you should also consider backup schedules. Generally, you should schedule backups to run after periodic stage-outs.

File Control Properties

File control properties influence, and in some cases determine, the selection of files to stage out. You can list file control properties and file sizes with **emls -l**. You can change file control properties with **emchmod**. The file control properties are listed in Table 11-7.

Table 11-7 **File Control Properties**

Property	Description
Locked	Locks the file onto magnetic disk; never stage out the file.
Convenient Stage Out	Stages out the file at the next convenient time, probably during the next periodic run.
Keep	Used in conjunction with convenient stage out to cause files to be prestaged rather than fully staged out.
Residence Priority	Prioritizes the importance of keeping the file on disk. All other things being equal, files with lowest priority are staged first. Priorities are expressed as integers. The highest priority is 1; the lowest priority is 63.

Directories have two sets of properties: one set applies to the directory itself (Directories are not staged out, so setting a directory's own properties has no effect), and the other is the inheritable set. When files are created, their own properties are inherited from the parent directory's inheritable set. When subdirectories are created, both the parent directory's own set and its inheritable set are inherited. Thus, file control properties are passed down through directory trees.

The residence priority remains set when a file is staged out and back in. The convenient and keep properties do not.

Although you should use all properties with care, be especially careful with the lock property. Choosing to lock some files on magnetic disk to increase the performance of one application could result in system-wide performance degradation. Locking too many files can prevent migration from working at all. Before locking files onto magnetic disk, try small changes to the residence priority. Monitor the system carefully to determine the effect both on the application and on the system as a whole. Be careful about setting the inheritable lock property on a directory. All files and directories that are created below that point inherit the lock property.

Listing and Changing File Control Properties

You can list file control properties and file sizes with **emls -l** (see Table 11-8 for flag names). The following example lists file control properties and file sizes for all of the files in the archive directory:

```
edm% emls -l archive
```

```
Mag KB Stg KB   I-flags      Flags     Staging media    Volume bercodes Filename
  1024     0    ----    0    1---  60        -                         filexyz
    24   898    ----    0    ----   0   #002-a  Archive                  fileabc
     1     0    --CK   60    ----   0        -                          dirabc
```

In this example:

- The file "filexyz" uses 1024 KBs of magnetic disk space and is locked on the magnetic disk.
- The file "fileabc" was staged out to volume #002-a on staging trail Archive, where it uses 898 KBs of disk space.

  A fencepost of 24 KB remains on magnetic disk. (A *fencepost* is the portion of a staged-out file that remains on magnetic disk.)
- Any files created in the "dirabc" directory are prestaged at migration's earliest convenience.

Directories have a set of inheritable properties, which are displayed in uppercase letters in the *I-flags* column. Both regular files and directories have a set of staging control properties that apply to the file or the directory; these properties appear in lowercase letters in the *Flags* column.

Table 11-8 **File Control Property**

Property	I-Flags	Flags
Locked	L	l
Convenient Stage-out	C	c
Keep	K	k

The *I-flags* column also displays the residence priority integer, which is a value of 1–31 (set by root) and 32–63 (set by ordinary users).

The *Flags* column displays the file or directory's own properties as the corresponding lowercase letters and priority integer. A minus sign (–) indicates that the property is not set.

You can change file control properties with **emchmod**. The **emchmod** command sets the file or directory's staging control properties. In order to change properties you must be either the superuser or the owner of the file. Unlike **chmod**, **emchmod** clears properties if they are not specified on the command line.

The following example shows how properties are inherited:

1. Assign convenient property and residence priority.
   ```
   edm# mkdir archive
   edm# emchmod -C -P36 archive
   ```
2. Display properties.
   ```
   edm# emls -l archive
   ```

```
Mag KB Stg KB I-flags  Flags   Staging media Volume Barcodes Filename
     1      0 --C- 36   ---- 0  -                            archive
```

3. Create subdirectories.

```
edm# cd archive
edm# mkdir arc1
edm# mkdir arc2
```

4. Display properties.

```
edm# emls -l *
```

```
Mag KB Stg KB I-flags  Flags   Staging media Volume Barcodes Filename
     1      0  --C- 36  ---- 0  -                             arc1
     1      0  --C- 36  ---- 0  -                             arc2
```

5. Create files.

```
edm# cd arc1
edm# touch file1
edm# touch file2
```

6. Display properties.

```
edm# emls -l *
```

```
Mag KB Stg KB I-flags  Flags   Staging media Volume Barcodes Filename
     0      0  ---- 0   --c- 36  -                             file1
     0      0  ---- 0   --c- 36  -                             file2
```

emchmod has several optional switches that allow you to expand the HSM file control properties. By default, symbolic links are not followed by **emchmod**. If you use the **-s** option, the change applies to all the symbolically linked files.

If you know when you create a directory what properties all the associated files should have, specify the inheritable properties at that time. If you decide *after* you have created a directory what properties it should have, use the **-r** option, to recursively apply the properties.

Normally, **emchmod** silently sets and or changes properties. If you use the **-v** option, **emchmod** prints a message to your screen as it changes every file.

Compaction of Staging Media

Over time, some files that were staged out are deleted. Other files are staged back in, and some of them are modified. The old staged image of a deleted or modified file is considered *stale*.

Gradually, the number of stale files on staging volumes grows, and the volumes become candidates for compaction. When you compact staging volumes, you actually stage in the files that are not stale to magnetic disk and then, if they were not accessed recently, you stage them out again to new staging volumes.

Compaction is in effect a garbage collection process that creates space for new files by reducing the number of active staging volumes. It frees space in a full library unit for new staging volumes and/or ensures a pool of available media.

In most cases, staging media is compacted automatically via an **emcompact** entry in root's crontab file. With automatic compaction, **emcompact** automatically determines which volumes to compact.

You can also compact staging volumes manually, if you want to compact any additional volumes. Both automatic compaction and manual compaction use the **emcompact** command.

If you need to compact some staging volumes manually:

1. Use **dbreport**'s compaction report to decide which volumes to compact.
   ```
   # dbreport compaction
   ```
   The compaction report is divided into three sections. The most likely volumes to compact are those listed in the last few lines of the first section. These are the volumes with the highest percentage of stale files.
2. Use **emcompact** to compact the volumes. In the case of EOs, which have two sides, you need to specify each side, or volume, separately. You can specify volumes by:
   - volume ID

- sequence number (for single-sided media)
- sequence number and side (for double-sided media)
- barcode (for single-sided media)

The following example compacts both sides of disk #10:

```
# emcompact EO 10-1 10-2
```

You can type the command without any arguments to find out the legal media types. (In the case of tapes, you can specify a barcode.)

You can override EDM Migration's file residence policy by running **emcompact** with the **–p** (policy) option. The **-p** option ensures that all files from the compaction source volume are staged out to the compaction output volume and none remain on magnetic disks.

Administering Compaction

You can perform several tasks on a regular basis to ensure that compaction is working smoothly.

CAUTION: **In order to ensure a complete file recovery process, you should disable automatic compaction and emvck as soon as you realize that you've lost a filesystem or a significant portion of one.**

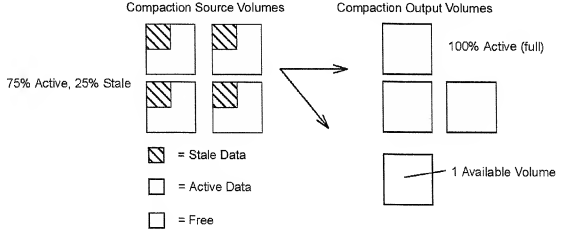
- Check the Volume Request window every morning to see if automatic compaction has blocked.
- Keep a supply of blank, easily accessible and unlabeled volumes, which you can label and allocate as compaction output volumes.

  To increase the likelihood of maintaining a supply of free volumes, you can also convert unrestricted staging trails to restricted ones, or prelabel volumes for a specific trail.
- Check the message files every day.

If automatic compaction frequently fails to reach the free goal, the system might have reached the limit of its data storage, given the number of volumes in the library unit or the time available for compaction.

In such a case, use the **dbreport appl_usage** report to determine whether enough stale space is available to reclaim, or whether you should add more volumes to your library unit. For example, in an EO system with volumes that average 25% stale data, you have to compact four volumes in order to get a single free volume (refer to Figure 11-10). If this rate is unacceptable in your environment, add more blank volumes.

Figure 11-10 **Freeing Volumes for Use**



Compacting Baseline Media

Compacting baseline volumes is similar to compacting staging volumes, except you can only compact baseline volumes manually. Compacting baseline volumes causes all currently active data associated with the volume to be copied to the current compaction volume associated with the baseline trail. New baseline compaction volumes are allocated as necessary.

As a general goal, you should limit the number of active baseline volumes to the number of active staging volumes. If possible, you should also limit the number of active baseline volumes to the number of slots in your library units. This facilitates recovery in the event of a site disaster by ensuring that all of the baseline media (to which active files are attached) fit in your library units.

To compact baseline volumes:

1. Run **dbreport baseline** weekly and refer to the "pct_stal" column for baseline volumes.
   ```
   # dbreport baseline
   ```
2. Use **emcompact** to compact the *N* most stale optical disks or tapes. (In the case of optical disks, there are two volumes per disk.) The value of *N* varies, but it is at least the number of active baseline media minus the number of active staging media or slots in your library units, whichever is less.

   Therefore, if you have 55 active baseline media, 45 active staging media, and a 50-slot library unit, compact at minimum the ten most stale baseline media.

   This results in a set of compacted baseline volumes. Note that these volumes are not deallocated and available for use until all existing baseline-relative backups that reference them expire.

Migration Reports

EDM HSM Option and EDM Migration contain several reporting tools that you can use to monitor system performance.

The **dbreport compaction** command generates three reports that you can use to determine which staging volumes to compact with the **emcompact** command.

The **emfsreport** program provides virtual filesystem statistics for the server and for network clients. It displays the amount of stageable and unstageable filesystem data, and the amount of data currently staged. Most importantly, it shows you the number of days worth of data being cached on magnetic disk. See "emfsreport and the Working Set" below for further details.

The **emsstat** program displays activity levels for the network migration server. It gets its information by accessing statistics that are kept in a shared memory segment that all active EDM Migration server daemon (emsd) processes use, or from a statistics file if emsd is not running.

The **emcheck** program checks the current migration configuration on the server or on a network client.

Refer to the appropriate man page for further details.

emfsreport and the Working Set

In a virtual filesystem, the files that you use most often, your *working set*, should fit on the magnetic portion of the filesystem, thus guaranteeing that most file accesses do not require staging volumes to be mounted. Your working set is often measured as the number of days worth of files that are stored on the magnetic disk. This is called the *working-set-in-days*. The ideal working-set-in-days varies from site to site and from filesystem to filesystem, but EMC recommends you try to maintain several days worth of data on your local magnetic disks.

Note that in the case of an archival environment where you expect to have no working set on your local disk, all accesses result in stage-in faults.

If your working set of files is considerably larger than your actual magnetic disk space, you experience system performance degradation. If you access large amounts of data within a short period, migration must stage files in and out continuously. This constant staging of files is called thrashing and should be avoided.

To find out your working set size and working-set-in-days, use **emfsreport**. The **emfsreport** program provides you with filesystem reports that enable you to monitor usage and thus fine tune your system. The **emfsreport** program can provide you with information such as:

- the number of files in a virtual filesystem
- the amount of stageable data in a virtual filesystem
- a filesystem's usage pattern for a particular day, for example, the total amount of space used by files created or modified in a day (that is, what is required for the green zone)
- the size of your working set

When you run **emfsreport** with the **-hva** option, you see a report of the virtual space by age since the last file access or modification. It also reports the size of the filesystem's working set. The working set *size* is the amount of stageable magnetic information that the filesystem can have without exceeding the LWM. The *days worth* value, which is based on observed access patterns, is the number of days it takes EDM Migration to cycle through an amount of data equal to the working set size.

Remember that this report is a snapshot of a specific moment in time. If you were suddenly to request many more files than normal, or if you were to create a large amount of new data, the working set period would be less than this estimate.

If you find that your working set is much larger than your physical space, that is, you have too few days worth of space, delete files or move them to another filesystem. Also, check that your system activity is evenly balanced across your disks. If your working set is still too large, add more magnetic disks or modify your application.

Baseline Backup

Baseline backup provides a highly efficient means of backing up large amounts of data. With baseline backups, you back up all of your most stable files, which, at minimum, consist of all the files that are staged out to the staging media. From that point on, you perform backups *relative* to the baseline; that is, the baseline backups take care of the data that is staged out, while the regular backups take care of everything else.

Baseline backup actually uses HSM software, rather than backup, to move data. In essence, it causes data to be staged out twice, and thus provides you with additional protection against the loss of your data. If, for example, you lose your primary staging media, due to fire or accident, you can still locate your files on the secondary staging media (that is, the baseline backups).

Restaging Data

HSM supports multi-level staging with its **restage** command, which incorporates enhanced **find** syntax. Using **restage** you can qualify files to stage and then migrate, or re-migrate files to a specified staging trail, force migration of a set of files, or establish an arbitrarily layered, staging hierarchy.

Multi-level staging is particularly useful when you want to free up space on your staging media. You can configure staging to automatically migrate data from magnetic disk to a staging trail, and then **restage** to another trail. If you want, you can move the restaged data to offline storage. See the **restage** man pages for more information.

Backup Completeness

Backup work items for filesystems that are under migration control have a completeness setting that prevents duplicate backups of the file data. The completeness setting limits the files for which the data portion is written to the backup. (The *extended inode* is included for each file scanned, regardless of whether its data is written out.)

You should leave the completeness settings at their defaults, listed in Table 11-9. The initial setting varies depending on the type of file being backed up (as noted in the table).

If you really need to change a completeness setting, you must edit eb.cfg directly; no setting is available from the graphical user interface. Editing eb.cfg directly is always a dangerous thing to do, so you should make a copy of your eb.cfg before editing.

Table 11-9 Completeness Settings

Setting	Description	Applicable For	Default For
All files	Back up the data portion of all files in the filespec, regardless of where they're stored and whether they're baselined.	All clients (this is the only option available for backup clients that are not also EDM Migration clients)	Non-migration clients and backup's database files on the server
Resident files only [1]	Back up the data portion for only those files that are resident (local to) the client; or, for the server, that are stored on the magnetic disk.	EDM Migration clients Levels 0-9 on the backup server (i.e., the local client)	—

Table 11-9 Completeness Settings (Continued)

Setting	Description	Applicable For	Default For
Files not backed up in migration store	If a file has been staged, only back up the data portion of the file if its staged version hasn't been backed up yet on the EDM Migration server.	EDM Migration clients	EDM Migration clients
Non-baselined files only	Back up the data portion for only those files that aren't baselined (for use after a baseline is taken). This option is only available if you have Baseline Backup.	Local (server) client (but not backup's database files)	Local client (except backup's database files)

1. EMC recommends that you use **Files not backed up in migration store** with EDM Migration network clients, and **Non-baselined files only** for the local migration server. The **Resident files only** setting can leave you vulnerable unless there is a backup of the client store. If you don't back up a file that has been staged out, but you lose the file's client store on the server before the server's files are backed up, the only way you will be able to recover the file is from an old backup.

12 How Migration Works

This chapter describes the roles of the HSM daemons, processes, and database files in migration services on the server and clients.

The following topics are discussed in this chapter:

- What Happens When You Enable Migration
- How Stage-Out Works
- How Stage-In Works
- How the User-Level Commands Work
- How the Network Migration Server Works
- How Compaction Works

What Happens When You Enable Migration

When you enable filesystem migration, you set up migration parameters, such as watermarks and a delay factor, and you specify the type of media to which files migrate.

When you enable filesystem migration, HSM does the following:

- It stores configuration information in the HSM configuration database.
- It creates a holding place for the migration candidate list (See "Candidate List Generation" on page 12-5).

Migration Configuration Database

The HSM server and every HSM client contains a migration configuration database. This database consists of structured text files that are updated by the **emstconf**, **emfsconf**, and **emsysconf** commands and by the functions you perform when using the HSM Configuration Interface.

> CAUTION: **Although these files are text files, you should never attempt to modify them with an ordinary editor. The configuration commands and the HSM Configuration interface do more than just modify the files; they also know how to interact correctly with any staging processes that are running.**

The database contains information that specifies which filesystems are stageable, when files should be staged, and which staging templates filesystems are assigned to.

On client systems, the database also lists the fileserver and store that staging templates are assigned to.

The database files are stored in the /usr/epoch/etc/mal/ directory.

Figure 12-1 **Configuration Database**

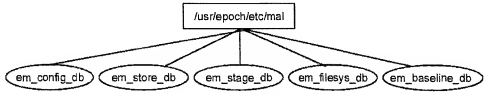


Table 12-1 lists the database files.

Table 12-1 **Migration Database Files**

Argument	Description
em_config_db	Text file containing system-wide configuration data. You update this file with the **emsysconf** command or when you change global properties with the HSM Configuration interface.
em_stage_db	Text file containing staging template configuration data. You update this file with the **emstconf** command or when you change staging template information with the HSM Configuration interface.
em_store_db	Text file containing client store information. You update this file with the **emstconf** command or when you change store information with the HSM Configuration interface.
em_filesys_db	Text file containing per-filesystem configuration data. You update this file when you issue the **emfsconf** command or when you change filesystem information with the HSM Configuration interface.
em_baseline_db	Text file containing baseline backup information.

How Stage-Out Works

HSM stages out files during nightly staging runs (periodic staging), when disk space usage crosses the high watermark (demand staging), or when the **emstage** command is issued. (See "When Files Stage In and Out" on page 11-2 for further details.)

There is a limitation of a maximum of 2GB minus 1KB for the size of files to be staged out or staged in.

Both periodic and demand stage-outs occur via the interaction of the following daemons and processes:

- The EDM Migration file monitor daemon (**emfmd**)
- The **emmasterd** daemon
- The **em_make_cl** process

Both the **emfmd** and the **emmasterd** daemon are started at boot time. The **emfmd** detects high watermark faults and then communicates with the **emmasterd** daemon, which starts a worker daemon. The worker daemon starts up **em_make_cl**, which fills in the candidate list and thus, determines which files to stage out.

These daemons and processes are described more fully on the following pages. See "How the User-Level Commands Work" on page 12-7 for information about **emstage**.

The File Monitor Daemon (emfmd)

The **emfmd** detects events that require migration intervention and then communicates with the **emmasterd** daemon. The **emmasterd** starts a worker daemon which actually stages out the file(s). (Previous versions of HSM provided the functions of the **emfmd** via the Unix kernel.)

The **emfmd** detects the following types of events:

- Filesystem space utilization at (or above) the high watermark

- Read or write accesses to prestaged or staged files
- Deletions of prestaged or staged files
- Filesystem mount and unmount operations

When a user program requests a file that is not staged out, the **emfmd** determines that no staging actions are necessary and normal system processing proceeds. The **emmasterd** daemon only gets involved when it is necessary to stage out a file.

The Master Staging Daemon (emmasterd)

When an HSM server or a migration client boots, it starts the master staging daemon, **emmasterd**, from /etc/rc3.d/S21mal. This daemon is responsible for staging out files from the clients to the server and from the server to the staging media. The **emmasterd** daemon keeps disk space utilization below the high watermark by staging out files and releasing their magnetic blocks.

Thereafter, the master starts, monitors, and restarts one worker daemon per filesystem, both periodically and on demand, when the **emfmd** notifies it that filesystem utilization exceeds the high watermark. The workers are also named **emmasterd**.

Only one real **emmasterd** process can ever run – the worker processes are simply forked copies. They appear as **emmasterd** processes when you run the **ps** command.

The simplest way to tell the difference between a worker process and the real **emmasterd** process is to look at the /usr/epoch/etc/mal/emmasterd.pid file. When **emmasterd** first begins execution, it writes its process ID into this file.

Candidate List Generation

When migration needs to stage files, an **emmasterd** worker process spawns **em_make_cl**, which creates a prioritized list of stageable files.

In selecting files to stage out, **em_make_cl** evaluates the time since the last file access, the size of the file, and the file's residence priority attribute (see the man page for **emchmod -p**).

Files with lower residence priority are usually staged first. (The lowest priority is 63; the highest priority is 1.) Thus, files with priorities from 33 to 63 are more likely to be staged out, and files with priorities from 0 to 31 are less likely to be staged out.

Only the superuser can raise priorities (by setting priorities in the range 1–31). All users can lower priorities.

What Happens When a File is Staged Out

The first time a file is staged out, migration writes the file's entire magnetic image to the next level in the staging hierarchy, that is, to the client store or the staging media. It keeps a small portion of the file on magnetic disk and releases the rest of the magnetic space. The portion of a staged-out file that remains on magnetic disk is called the *fencepost*.

This fencepost is useful, because many commands, such as **file** and **head**, only need to read this small portion of the file. Consequently, when these commands are run, HSM doesn't need to stage in the entire file. When a file is staged out a second time, it releases the magnetic space occupied by the fencepost.

Files that are staged in reside on both magnetic disk and the staging media (or client store). If the file is staged out again without being modified, migration uses the same staged image and releases the magnetic space. If the file is modified and then staged out, migration writes a new image on the staging media or client store.

How Stage-In Works

Stage-ins occur due to the interaction of the HSM file monitor daemon (**emfmd**) and the stage-in daemon (**emsid**). The **emfmd** detects a request for a staged out file (or a request to delete a staged out file) and notifies the **emsid**, which stages in some portion of the file, or, in the case of delete operations, deletes the file's staged image.

Scripts that run at system boot time automatically start up several stage-in daemons. Each stage-in daemon can handle one stage-in request at a time, which allows for multiple, simultaneous stage-in requests.

A staged-out file is logically divided into small chunks, called *buckets*. A file can be divided into anywhere from one up to a maximum of 64 buckets. The size of the buckets is based on the file's size. As the size of the file increases, the size of the bucket also increases.

When certain applications and processes request to read a portion of a staged-out file, migration stages in only those buckets that contain the requested data. Whenever a file is modified in any way, migration stages in the entire file and makes the previous staged image invalid or *stale*.

How the User-Level Commands Work

The user-level commands (**emchmod**, **emls**, **emstage**, and **embsi**) enable users to set and list file attributes for stageable filesystems and to specifically request the staging of particular files. The user-level commands interact with the migration RPC daemon (**emrpcd**), which, in turn, interacts with the **emfmd**.

For more information about the commands, see the man page for each individual command.

How the Network Migration Server Works

Network migration server software runs on the EDM server and consists of the following components as listed in Table 12-2.

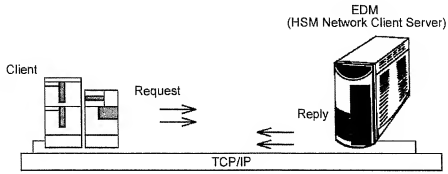
Table 12-2 **Components of the Network Migration Server**

Component	Description
EDM Migration protocol	The communication between server and clients
EDM Migration daemons	Daemons that service client requests
Network migration server database	Files that track network migration activity and the default client store values
Client stores	Directories that hold client bitfiles

The EDM Migration Protocol

Network migration server software uses the EDM Migration protocol to enable communication between the clients and the server. The EDM Migration protocol is a remote procedure call (RPC) protocol that consists of pairs of request and reply messages that are passed between the client and server.

Figure 12-2 **EDM Migration Protocol**



The EDM Migration protocol is based on a connection-oriented transport protocol (TCP/IP), requiring each client to establish one or more virtual circuit connections to the server. This protocol reduces the effects of transport latency (round trip time) on performance and permits network migration to function well over both local and wide area networks.

Network Migration Server Daemons

Network migration server activities are carried out by a hierarchy of daemons (all named the EDM Migration Server Daemon, **emsd**) and controlled by a set of administrative commands. (See the appropriate man pages.)

The **emsd** is responsible for the following activities:

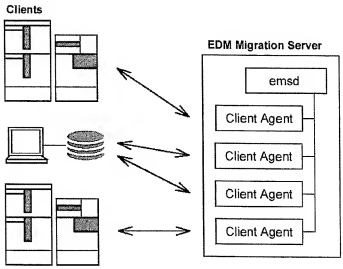
- initializing the global statistics shared memory segment
- parsing the global configuration file and store database
- parsing the store configuration files for each client store
- registering Network RPC service information
- listening for client system connection requests

The **emsd** daemon is started at boot time. When client connection requests arrive, **emsd** spawns subprocesses called *client agents* to handle them. The **emsd** creates a client agent process for each connection.

The client agent handles all requests over that connection for the lifetime of the connection.

Figure 12-3

Server Daemons



When an agent receives a request to access bitfiles in a particular store, the agent looks up the store configuration and state information in data structures inherited from the emsd process. A client agent can access any store to which its client system has access permission.

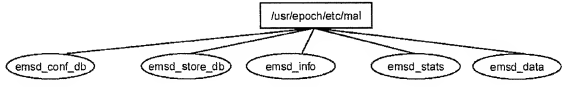
Network Migration Server Database

The network migration server has a global configuration database that contains information about network migration activity and the default client store values. The global configuration database files reside in /usr/epoch/etc/mal.

Although both the emsd_conf_db and the emsd_store_db files contain editable text descriptions of the configuration, do not edit these files directly. Instead use the server's configuration commands or the Configuration Interface to make any modifications to the database.

CAUTION: Editing these files directly may result in loss of data.

Figure 12-4 **Global Database Files**



There are five database files as described in Table 12-3.

Table 12-3 **Global Database Files**

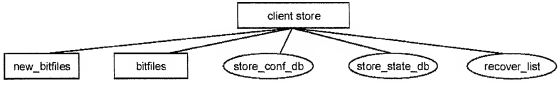
File	Description
emsd_conf_db	Text file that defines the limits on the EDM Migration protocol requests and the default values for client store configurations.
emsd_info	Binary file that contains information about the currently executing emsd process.
emsd_store_db	Text file that contains a list of configured client stores and their locations on the server.
emsd_stats	Binary file that contains cumulative statistics on EDM Migration protocol traffic and client agent activity.
emsd_data	Binary file that contains the EDM Migration usage history on the server.

Client Stores

Each client store has its own file hierarchy and is logically independent from every other client store. The client store's top-level directory contains three files and two subdirectories.

As system administrator you see these files and directories when you list the contents of the client store directory.

Figure 12-5 **Client Store Organization**



The client store's top-level files and directories are listed in Table 12-4.

Table 12-4 **Client Store Files and Directories**

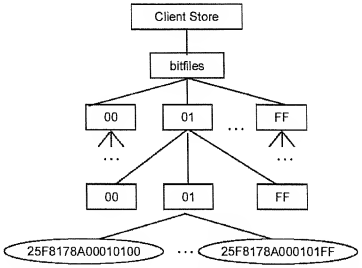
File/Directory	Description
store_conf_db	Text file of the store's configuration information.
store_state_db	Text file of the store's state information that the client agent keeps current.
recover_list	List of the bitfiles to restore from the server's backups.
new_bitfiles	Temporary holding directory for bitfiles that are being created as part of a stage out from a client system.
bitfiles	Directory that contains the completed bitfiles in a 3-level hierarchy.

When the client agent creates a bitfile, it gives it a 16-digit hexadecimal name and places it in the new_bitfiles directory. The bitfile remains there until it is completely written. Once the bitfile is complete, the client agent moves it to the bitfiles directory.

Bitfiles

Bitfiles are stored at the bottom layer of a directory hierarchy as shown in Figure 12-6. Bitfile names are 16-digit hexadecimal numbers representing the lower 64-bits of a file's bitfile ID. (The bitfile ID consists of the bitfile name plus the store ID.) Migration uses this organization so that a bitfile can be located by using the hexadecimal encoding of the bitfile ID.

Figure 12-6 **Bitfile Hierarchy**



How Compaction Works

Compaction is in effect a garbage collection process that creates space for new files by reducing the number of active staging volumes. On systems with erasable staging media, compaction occurs automatically via an **emcompact -c** entry in root's crontab file.

Compaction Goals

The object of automatic compaction is to reclaim storage space on the staging media by making sure that each staging trail has at least a certain number of available volumes to stage to. By always maintaining a sufficient pool of volumes, minimal operator intervention is needed.

The **emcompact** command operates, however, under a pair of competing goals:

- To compact as many volumes as possible without blocking itself. (A block occurs when EDM Migration needs to allocate a new volume in order to compact a volume, but there are no new volumes available.) The **emcompact** program will block while waiting for a new volume, and a new volume request will be posted to the Volume Request window.
- To begin compaction with the trail that needs it the most, that is, the trail with the fewest number of available volumes.

The **emcompact** program only considers the volumes in the library units to determine the number of volumes available for allocation to each staging trail. If that number is less than the value specified in the **-a** (automatic) option, it examines all the volumes and compacts enough to provide an adequate number of free volumes. By default, **emcompact** ensures there are at least three free volumes for each staging trail. After the source volumes are compacted, they are erased and made available for reuse.

Example

In the following crontab entry, **emcompact** is set to run at 1:00 a.m. every morning:

```
00 1 * * * PATH=/usr/epoch/bin:$PATH;export PATH;emcompact -c  >/dev/null 2>&1
```

This command specifies that autocompaction should be done according to the directives contained in the autocompaction configuration file /usr/epoch/etc/mal/emcompact.cfg. This form is typically called to compact reusable volumes in a library unit. See cron(1m) and the /usr/lib/crontab and /usr/epoch/etc/mal/emcompact.cfg files distributed with the system.

The Compaction Process

When you run compaction automatically, **emcompact** first chooses a staging trail and then decides which volumes to compact within that trail. For each staging trail, **emcompact** can allocate, as compaction output volumes, any volume that is already allocated to that staging trail or any unrestricted volume from another staging trail, as long as both sides are available.

Table 12-5 shows some sample staging trails and the volumes available for each trail at a certain point in time. This information is used in the process that is described below.

Table 12-5 **Available Compaction Volumes**

Staging Trail	Available Volumes
Engineering	6
Engineering_archive	4
Documentation	2
Documentation_archive	2
CAD	1
CAD-archive	0

The process is as follows:

1. First, **emcompact** looks for trails that have less than *n* available volumes. (*n* is the number specified with the **-a** option, 3 in this example). Automatic compaction only operates on those trails that have less than *n* available volumes, so **emcompact** selects only the Documentation, Documentation_archive, CAD, and CAD_archive staging trails.
2. Then, **emcompact** selects the trails that are least likely to cause a blocked process and require operator intervention. As such, it first looks for a trail with at least two available

volumes, in this case, Documentation and Documentation_archive. If there's more than one trail with at least two available volumes, but less than *n* available volumes, **emcompact** first chooses the trail with the fewest available volumes.

3. Then, **emcompact** selects the piece of media that is the most stale, taking into consideration *both* volumes, in the case of an optical disk. The **emcompact** program also takes into consideration the disks' availability, so that any disks that are restricted to other trails cannot be considered.
4. On that disk, **emcompact** selects the stalest side and compacts it.
5. The **emcompact** program repeats Step 2 through Step 5 until all the trails that started with at least 2 available volumes have had volumes freed up and now have at least *n* available volumes.
6. The **emcompact** program repeats Steps 2 through Step 5 until all the trails that started with 1 available volume (CAD) have at least *n* available volumes. At this point there is a greater potential for a blocked process.
7. The **emcompact** program repeats Steps 2 through Step 5 until all the trails that started with 0 available volumes have at least *n* available volumes. At this point there is the greatest potential for a blocked process.

At any time in this sequence, **emcompact** can run out of time, depending on the length of time specified in the **-e** switch. For example, if the command line specifies **-e 120**, **emcompact** will terminate in two hours. In this case, the program will exit. The next time compaction runs it starts the selection process over again. Most likely, it decides that the volume it was in the process of compacting is the best candidate to compact.

How Long Compaction Takes

Compaction takes a considerable amount of time; up to a few hours is not unusual. **emcompact** requires about five minutes to scan each filesystem and identify active files. The time required to stage the files in and out again depends on the number of blocks and can increase if compaction triggers event-driven staging. During compaction, all resident and staged-out files (including the files staged out to the volumes being compacted), in all filesystems, can be used normally. If compaction is interrupted, you can restart it on the same volumes.

Baseline Compaction

Compacting baseline volumes is similar to compacting staging volumes. Compacting baseline volumes causes all currently active data associated with the volume to be copied to the current compaction volume associated with the baseline trail. New baseline compaction volumes are allocated as necessary.

When that baseline volume is compacted it cannot be immediately reused since there may still be baseline-relative backups referencing the volume. When you compact a baseline volume you move the data of all active files that reference this volume to a different baseline volume. This means that no new baseline-relative backups reference this baseline volume.

A compacted baseline or staging volume has no active data. You can verify this by checking the volume's **Used 1k blocks** field (for an optical volume) or **Used files** field (for DLT) with the EDM Library Unit Manager window.

Deallocation and Reuse

The deallocation and reuse of baseline volumes is handled by EDM Backup. When expiring backups, EDM Backup checks for active data on all baseline volumes by checking each volume's "KB used" field. If the field is zero, either due to the volume having been compacted or the volume having grown completely stale, EDM Backup considers this volume for deallo-

cation. After all of the baseline-relative backups that reference these baseline volumes expire, **ebexpire** can deallocate the baseline volume, making it available for reuse.

Note that you can compact any baseline volume at any time. The deallocation of the volume is handled by EDM Backup, which knows not to deallocate it if it is still required by any baseline-relative backup.

Active Baseline Volumes

A baseline volume is considered "active" if it has data on it that is needed by an unexpired baseline-relative backup. Such a baseline volume cannot be reused until it has been compacted *and* deallocated.

Recovering from Site Disasters

You should limit the number of active baseline media to the number of active staging media, and you should make sure that you can fit all of your active baseline media into your library unit at one time.

The reason for this is that in the case of a site disaster you need to replace your damaged staging media with your baseline media. (Note, however, that although this should be a maintenance goal, there is no real relationship between staging and baseline media.)

Furthermore, in the case of a site disaster, the fewer baseline volumes you need to deal with, the better. That is, you do not want to have to purchase an extra library unit because you have more baseline media than staging media.

See "Compacting Baseline Media" on page 11-29 for information on how to compact baseline volumes.